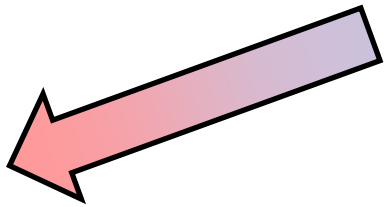
Unsupervised and Generative AI for High Energy Physics and Beyond

Kevin Pedro
(Fermilab)
November 14, 2024

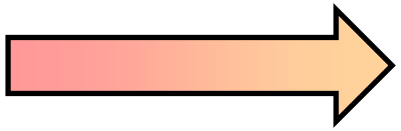
Outline

High Energy Physics, Multimessenger Astronomy, & AI/ML

Unsupervised Learning



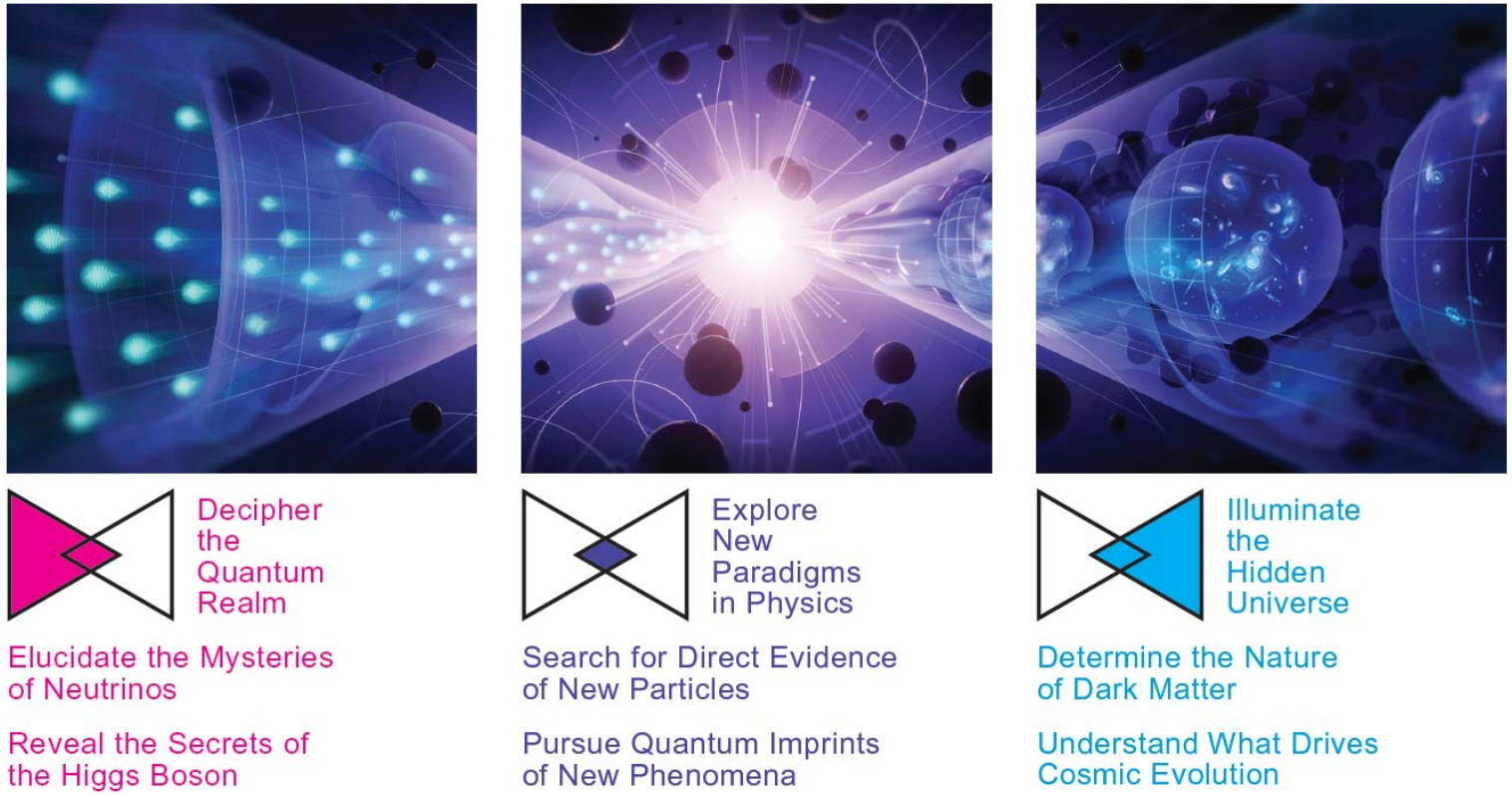
Generative Learning



Inference as a Service

High Energy Physics, Multimessenger Astronomy, & AI/ML

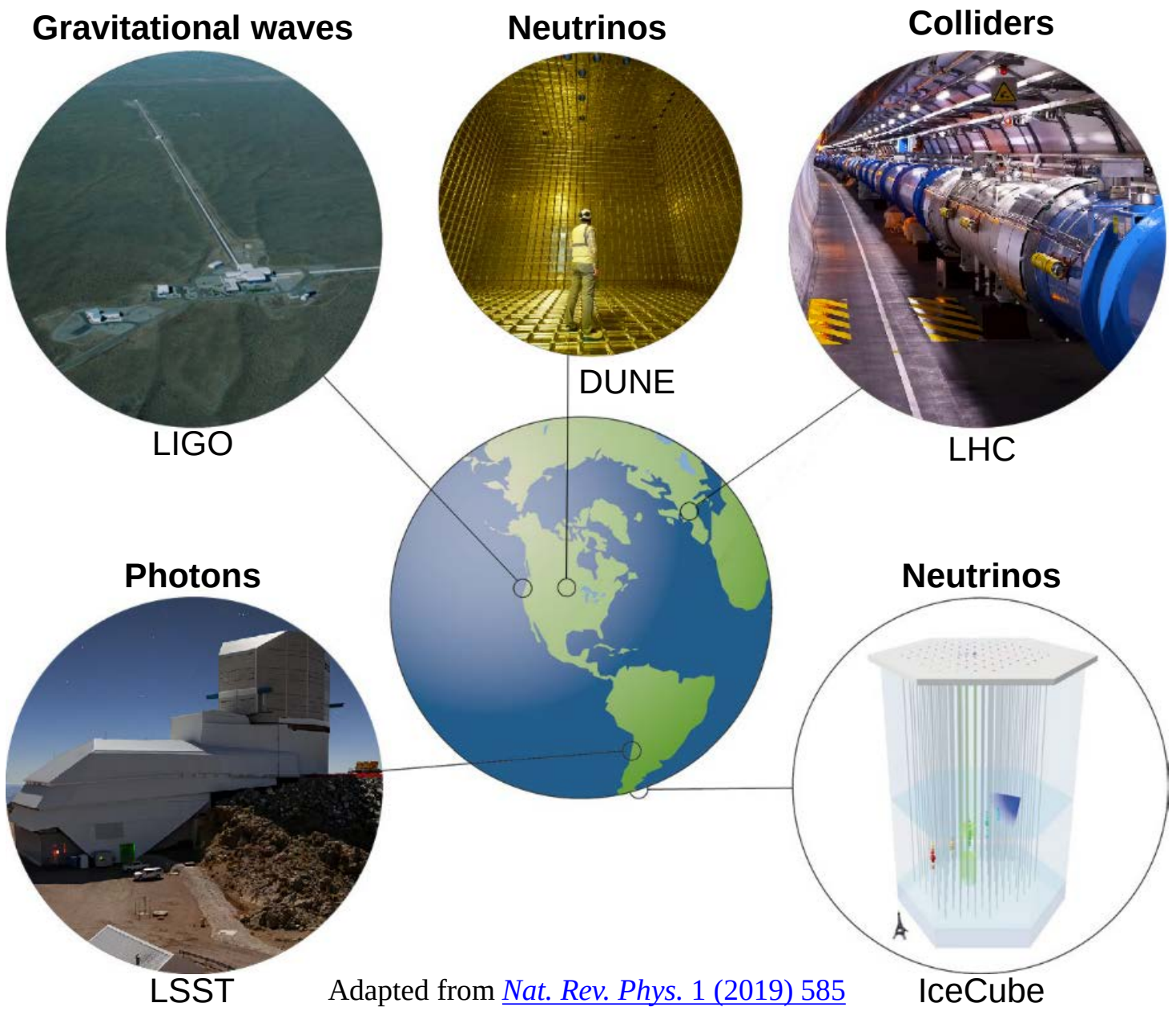
Exploring the Quantum Universe


Decipher the Quantum Realm
Elucidate the Mysteries of Neutrinos
Reveal the Secrets of the Higgs Boson
Explore New Paradigms in Physics
Search for Direct Evidence of New Particles
Pursue Quantum Imprints of New Phenomena
Illuminate the Hidden Universe
Determine the Nature of Dark Matter
Understand What Drives Cosmic Evolution

- New decadal plan for high-energy physics emphasizes complementarity with astrophysics
  - High-energy physics (HEP): direct manipulation and measurement of particles at the smallest scales
  - Multimessenger astronomy (MMA): observing the largest scales using particles as tools

Laser Interferometer Gravitational-Wave Observatory:
observe gravitational waves from black hole & neutron star mergers, supernovae

Legacy Survey of Space and Time (Rubin Observatory):
survey entire visible sky from near-ultraviolet to near-infrared wavelengths



Adapted from *Nat. Rev. Phys.* 1 (2019) 585

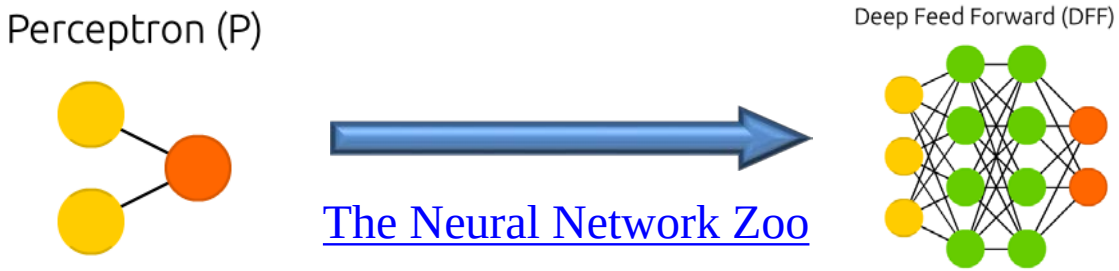
Large Hadron Collider:
collide protons at $\sqrt{s}$ = 13.6 TeV

IceCube and *Deep Underground Neutrino Experiment*:
observe and measure cosmic and accelerator neutrinos

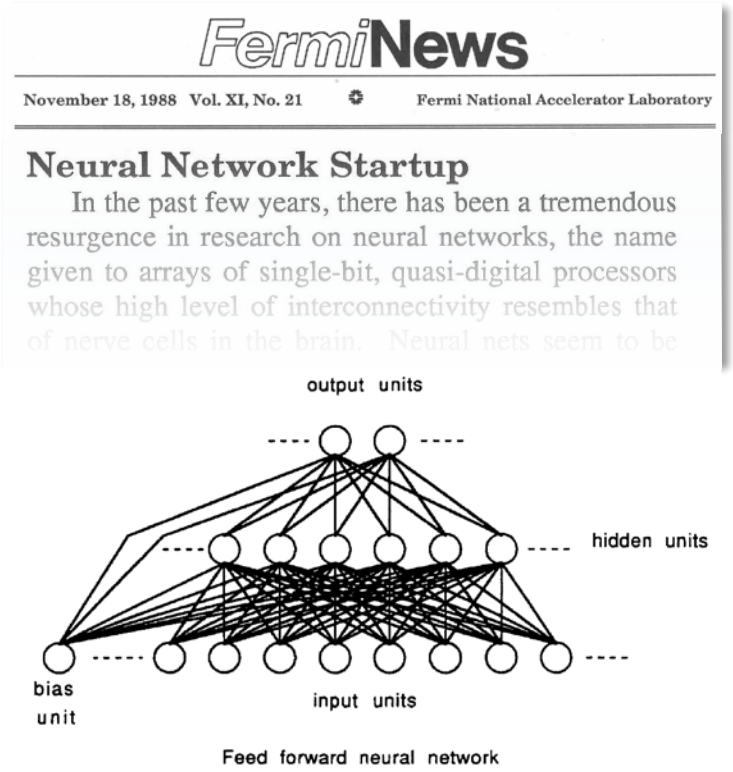
What is AI/ML?

> "AI is whatever hasn't been done yet."
> – Douglas Hofstadter

- ML is *function approximation*:
  - ➢ map inputs to outputs, $\vec{x} \mapsto \vec{y}$
  - o $\vec{y} = F(\vec{x})$ unknown, probably not analytic
    → try to find approximation $\vec{y} \approx F'(\vec{x}; \vec{w})$ by optimizing *weights* $\vec{w}$ (in general, any parameters)
- Deep learning:
  - o Use thousands, even millions of weights
  - o Use many *layers* with intermediate *features* derived from inputs
    - ▪ More "neurons" → more multiplications



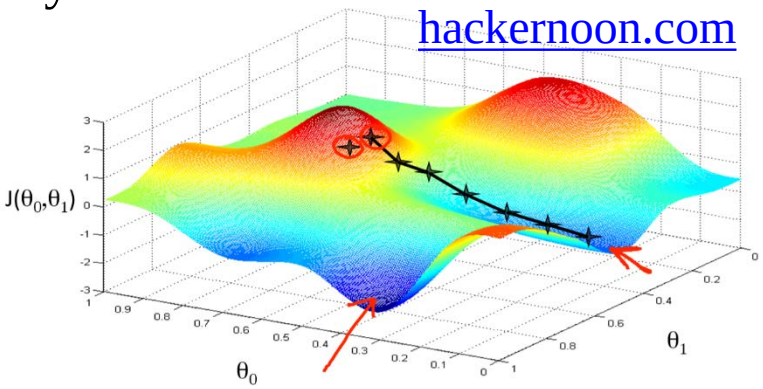
History of AI/ML in Physics


FermiNews
November 18, 1988 Vol. XI, No. 21
Fermi National Accelerator Laboratory
Neural Network Startup
In the past few years, there has been a tremendous resurgence in research on neural networks, the name given to arrays of single-bit, quasi-digital processors whose high level of interconnectivity resembles that of nerve cells in the brain. Neural nets seem to be
output units
hidden units
bias unit
input units
Feed forward neural network

[B. Denby, May 1990]

- Physics experiments provide ample input data $\vec{x}$
  - From low-level detector hits to high-level reconstructed objects
- AI training: iteratively modify weights so F′ gets "closer" to $\vec{y}$
  - "Closer" defined by a *loss function*
  - Use *gradient descent* to follow change in loss
- This is a *supervised* approach
  - Need to know and specify both $\vec{x}$ and $\vec{y}$
- Works well for tasks like:
  - Classification
    - What type of particle or galaxy?
  - Regression
    - What is the mass of this particle?
    - What is the redshift of this galaxy?
- Used for these questions for almost 40 years now

hackernoon.com



Future of AI/ML in Physics

- What if we don't know the answer?

➢ Learn *probability density* of inputs

  - Various approaches, w/ tradeoffs: usually speed vs. accuracy

- Commonly called "generative"

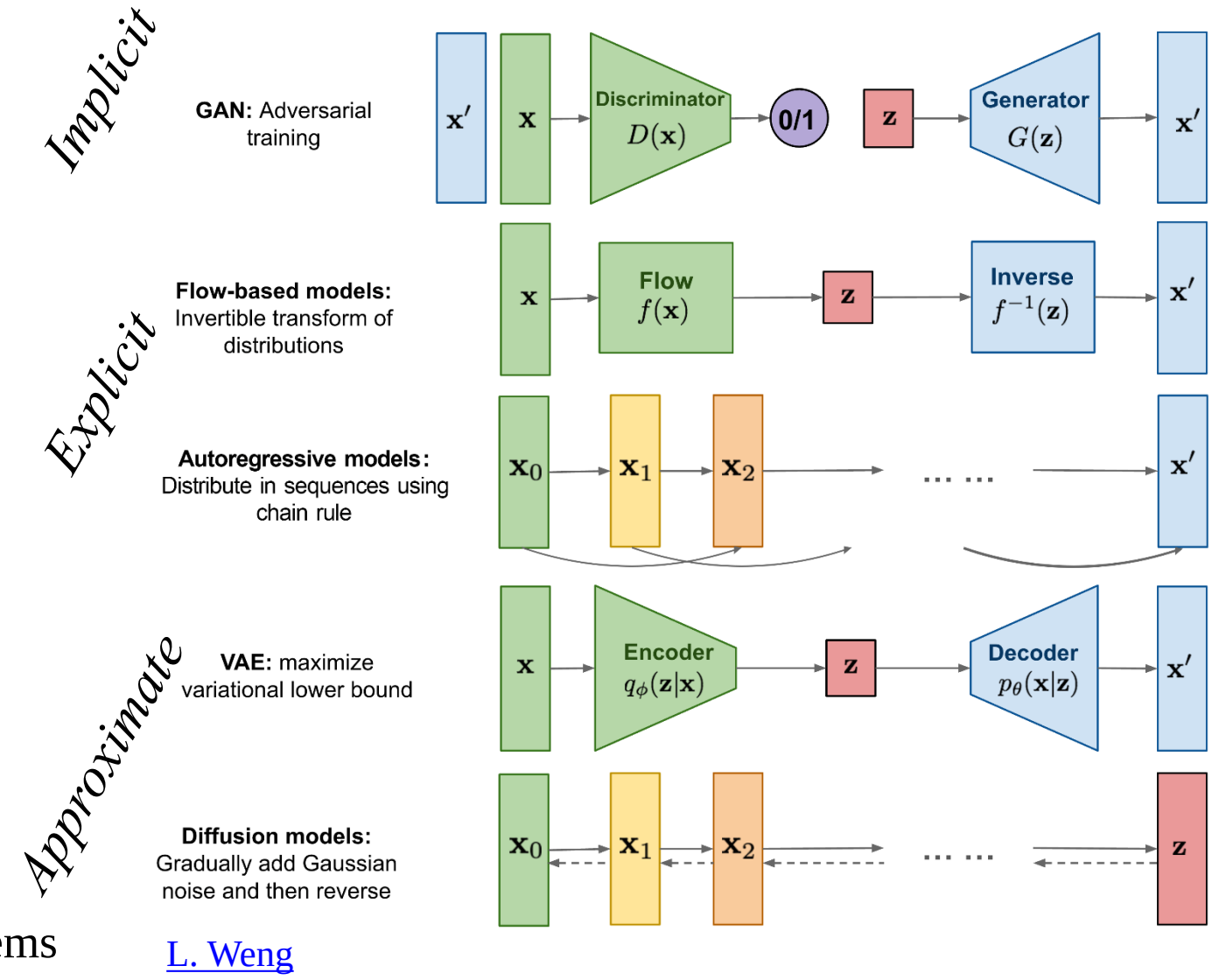
  - Learn density → generate new examples

- Also inherently "unsupervised"

  - $\vec{y}$ does not appear anywhere here!

  - Learning *about* the inputs themselves

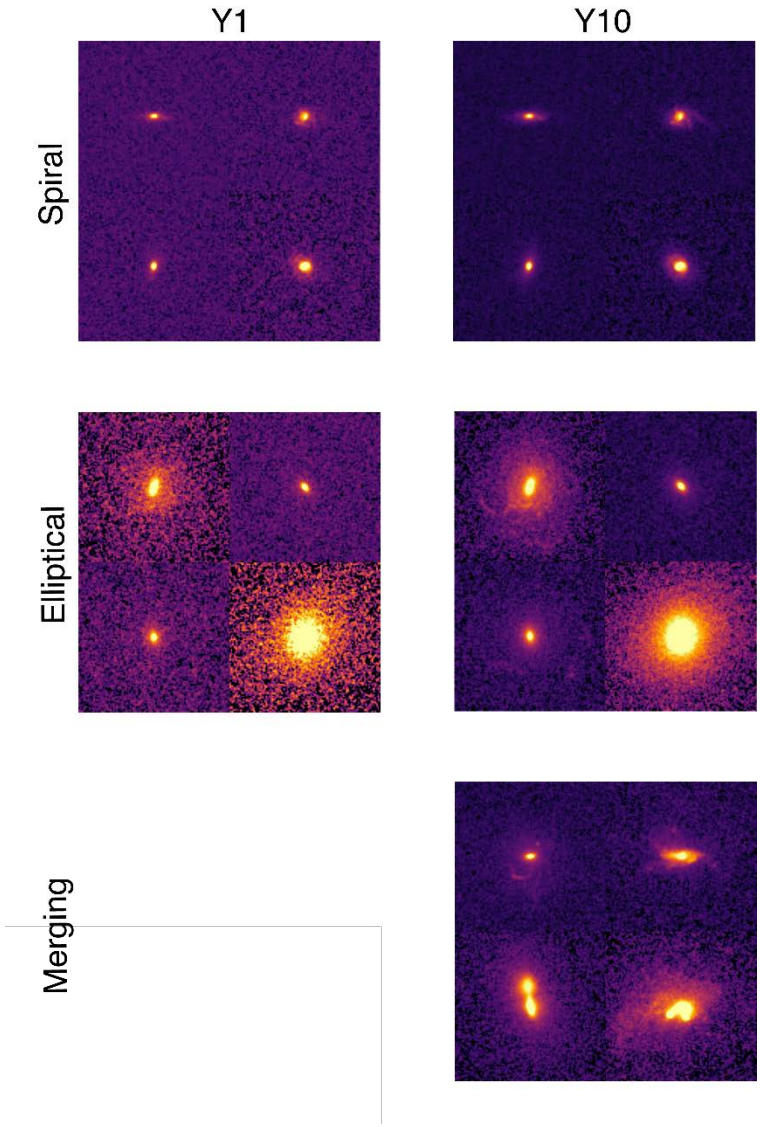
- Potential to solve whole new types of problems



L. Weng

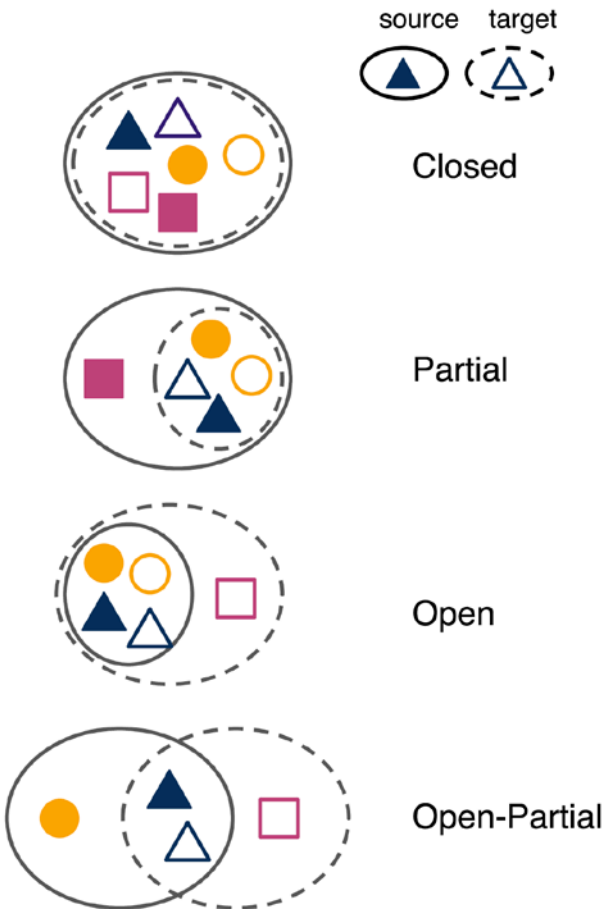
Unsupervised Learning

Galaxy Classification

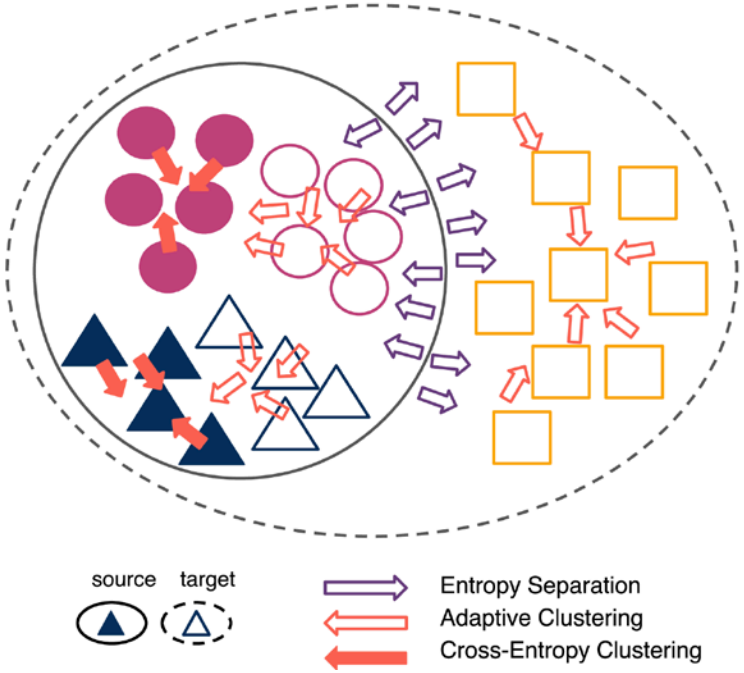


- Simulate what LSST will observe after 1 and 10 years of running
  - Noise decreases as observation time increases
- Frequencies of different types of galaxies important to understand cosmic evolution → need to classify
- Several problems with supervised classifiers:
  - Differences between simulated and real data may lead to false positives or negatives
  - Similarly for differences between early and late data, and worse: what if late data contains new classes not found in early data?
  - Consistency across surveys: want to learn underlying physics that distinguishes different galaxy types → sufficiently general algorithm should perform similarly on different datasets
    - Sloan Digital Sky Survey (SDSS), Dark Energy Camera Legacy Survey (DECaLS), etc.

Universal Domain Adaptation



- Domain adaptation:
  - Ensure consistent performance across multiple datasets
- Source domain (labeled) may differ from target domain (unlabeled) in multiple ways
  - Different properties, different classes, or both
- *Semi-/self-supervised*, *contrastive* approach handles all cases!



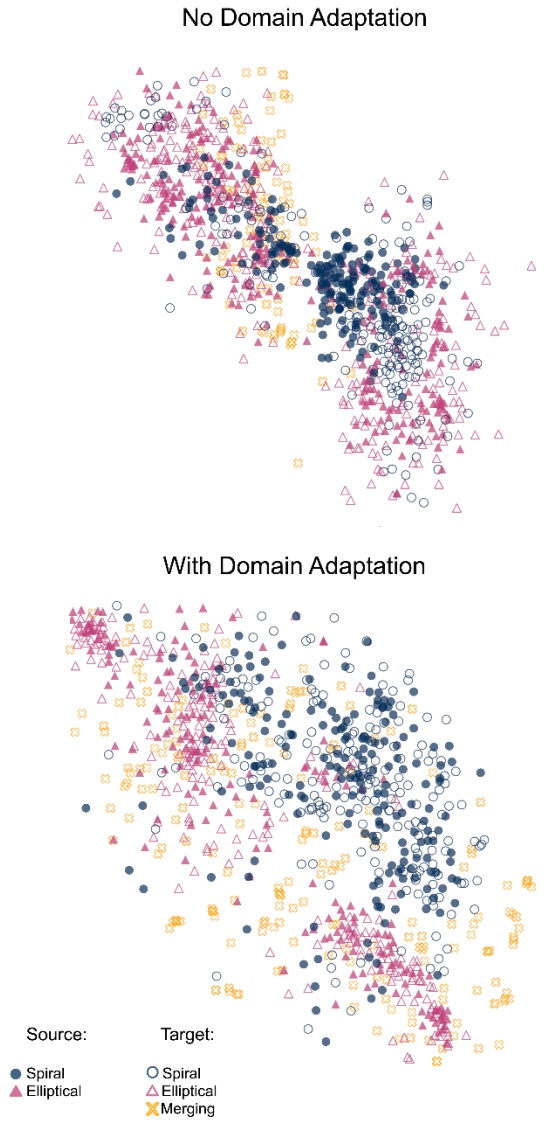
$$\mathcal{L}_{\mathrm{AC}} = -\sum_{i \in B} \sum_{j \in b_t} s_{ij} \log(\hat{s}_{ij}) + (1 - s_{ij}) \log(1 - \hat{s}_{ij})$$

s_{ij}: similarity score, true if same prediction ordering for known classes

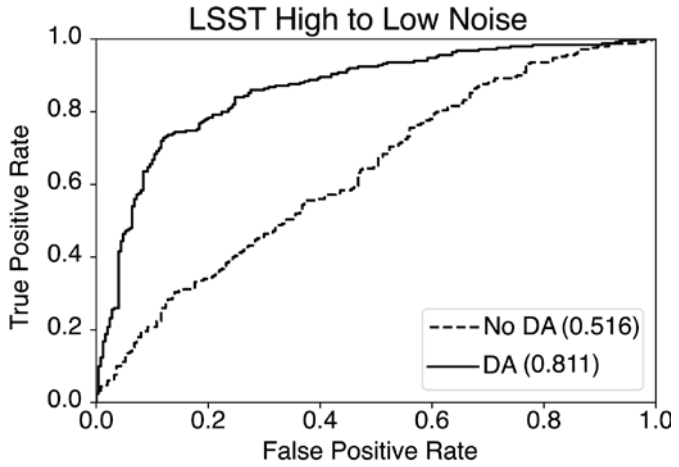
$$\mathcal{L}_{\mathrm{ES}}(\boldsymbol{p}_i) = \begin{cases} -|H(\boldsymbol{p}_i) - \rho| & |H(\boldsymbol{p}_i) - \rho| > m \\ 0 & \text{otherwise.} \end{cases}$$

ρ: entropy boundary,
m: confidence interval;
actively tuned during training

Cross-Time Results



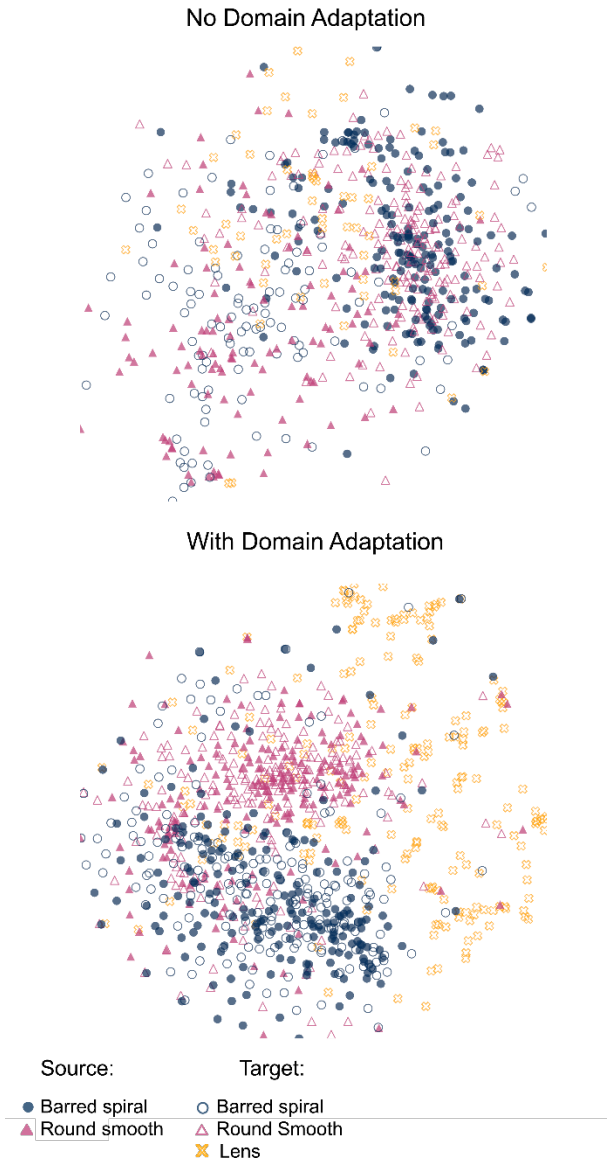
- Visualize latent space using combination of:
  - isomap (retain geometric distances in lower-dimensional representation)
  - t-distributed stochastic neighbor embedding (tSNE; ensure data pairs have same probabilities in lower-dimensional space)
- With DA:
  - Known classes become better aligned across domains
  - Unknown class separated from known classes



- Improvement on *all* metrics for *both* domains!

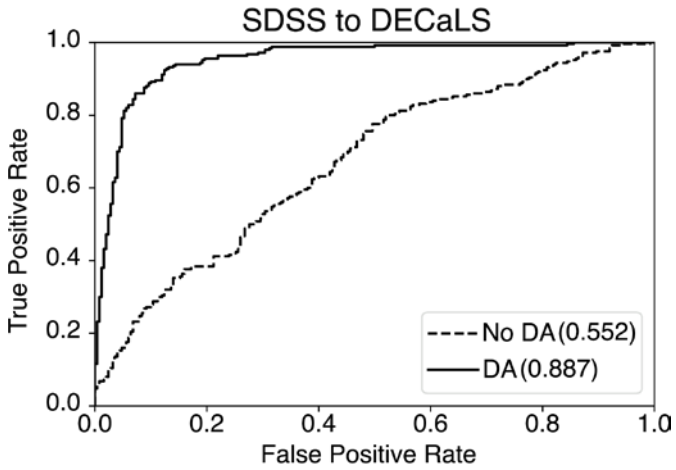
		High → Low	
Training	Metric	Source	Target
Reg.	Accuracy	0.61 ± 0.32	0.33 ± 0.48
	Precision	0.65 ± 0.34	0.31 ± 0.46
	Recall	0.54 ± 0.28	0.33 ± 0.39
	F1 Score	0.59 ± 0.22	0.32 ± 0.31
DA	Accuracy	0.70 ± 0.20	0.73 ± 0.18
	Precision	0.64 ± 0.30	0.72 ± 0.17
	Recall	0.66 ± 0.24	0.74 ± 0.20
	F1 Score	0.65 ± 0.19	0.73 ± 0.13

Cross-Survey Results



- Treat SDSS as labeled, DECaLS as unlabeled
  - Showing only 3 out of 10 classes
- Without DA:
  - Source and target anti-aligned
  - Unknown class mixed in everywhere
- With DA:
  - Source and target aligned
  - Unknown class well-separated
  - Improvements in all metrics

➢ UDA works for any sources, any targets, any number of classes



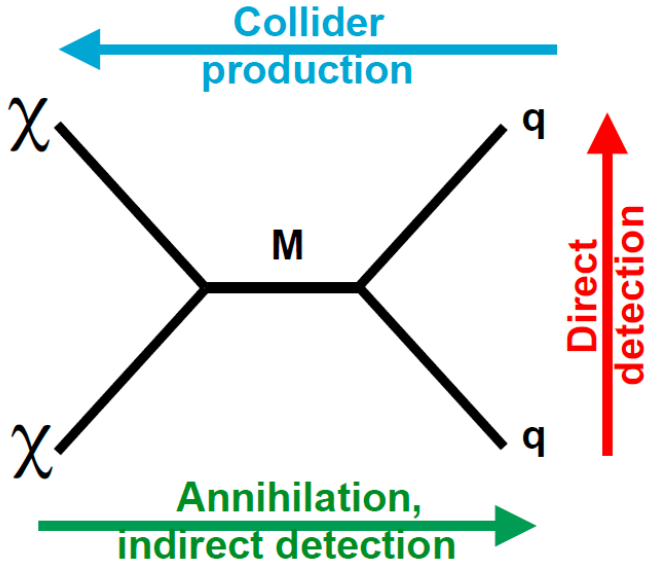
SDSS → DECaLS

Training	Metric	Source	Target
Reg.	Accuracy	0.77 ± 0.20	0.43 ± 0.32
	Precision	0.52 ± 0.23	0.42 ± 0.43
	Recall	0.70 ± 0.25	0.33 ± 0.30
	F1 Score	0.60 ± 0.18	0.37 ± 0.25
DA	Accuracy	0.82 ± 0.09	0.79 ± 0.10
	Precision	0.79 ± 0.06	0.73 ± 0.21
	Recall	0.84 ± 0.13	0.81 ± 0.18
	F1 Score	0.81 ± 0.07	0.77 ± 0.14

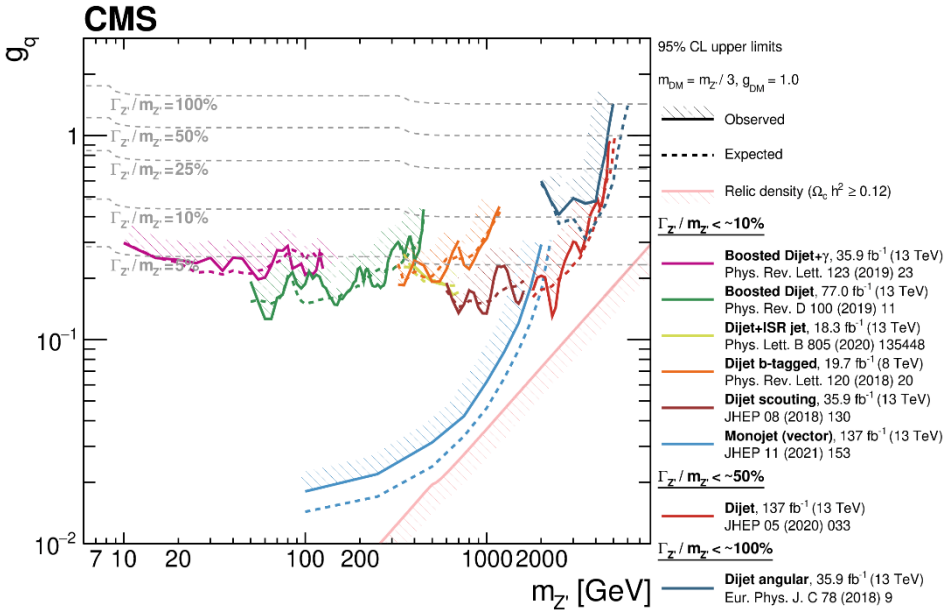
Dark Matter Searches

We know dark matter exists and behaves differently from visible matter

Collider, direct, and annihilation searches have largely focused on weakly interacting massive particle (WIMP) signatures

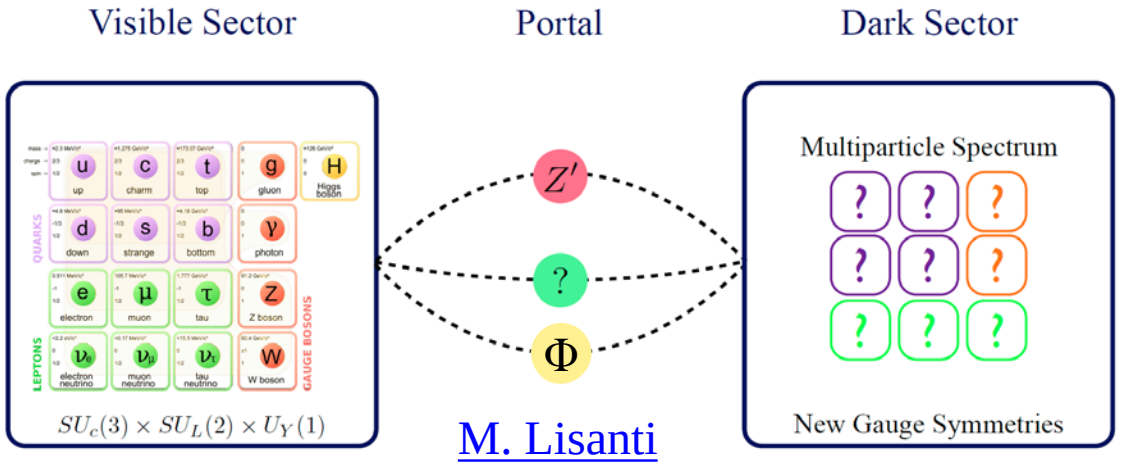


But so far, no direct experimental evidence of its nature


CMS
g_q
95% CL upper limits
$m_{DM} = m_{Z'} / 3$, $g_{DM} = 1.0$
$\Gamma_{Z'}/m_{Z'}$=100%
$\Gamma_{Z'}/m_{Z'}$=50%
$\Gamma_{Z'}/m_{Z'}$=25%
$\Gamma_{Z'}/m_{Z'}$=10%
$\Gamma_{Z'}/m_{Z'}$=5%
Observed
Expected
Relic density ($\Omega_c h^2 \geq 0.12$)
$\Gamma_{Z'}/m_{Z'}$ < ~10%
Boosted Dijet+γ, 35.9 fb^{-1} (13 TeV) Phys. Rev. Lett. 123 (2019) 23
Boosted Dijet, 77.0 fb^{-1} (13 TeV) Phys. Rev. D 100 (2019) 11
Dijet+ISR jet, 18.3 fb^{-1} (13 TeV) Phys. Lett. B 805 (2020) 135448
Dijet b-tagged, 19.7 fb^{-1} (8 TeV) Phys. Rev. Lett. 120 (2018) 20
Dijet scouting, 35.9 fb^{-1} (13 TeV) JHEP 08 (2018) 130
Monojet (vector), 137 fb^{-1} (13 TeV) JHEP 11 (2021) 153
$\Gamma_{Z'}/m_{Z'}$ < ~50%
Dijet, 137 fb^{-1} (13 TeV) JHEP 05 (2020) 033
$\Gamma_{Z'}/m_{Z'}$ < ~100%
Dijet angular, 35.9 fb^{-1} (13 TeV) Eur. Phys. J. C 78 (2018) 9
1
10^{-1}
10^{-2}
7 10
20
100
200
1000
2000
$m_{Z'}$ [GeV]

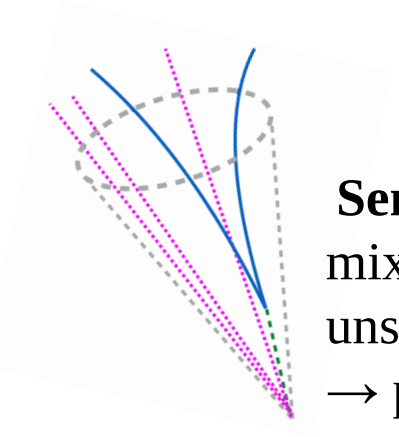
Dark QCD

- A *strongly coupled dark sector* with composite DM ("dark QCD") would evade WIMP bounds:
  - *Novel* collider phenomenology – ignored or rejected by typical strategies
  - DM abundance arises from asymmetry mechanism → no annihilation
  - DM interactions with ordinary matter highly suppressed → no direct detection
- Cosmological motivations:
  - Most visible matter is baryonic (composite); maybe DM is similar
  - DM density similar to SM density (~5× larger); $m_{DM} = 5m_{proton}$?
- Such models are extremely complicated with dozens of parameters:
  - How to conduct a broad search?



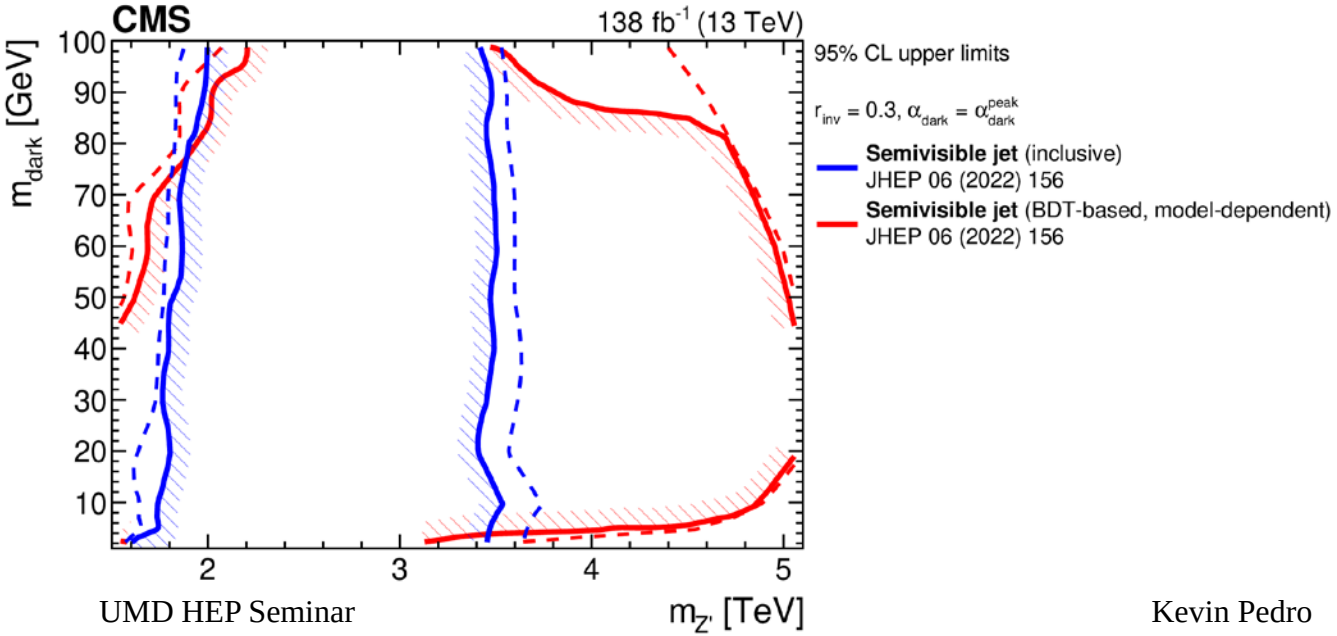
M. Lisanti

Supervised Learning for Dark QCD

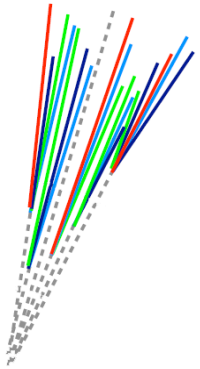


Semivisible jets (SVJs)
mixture of stable and unstable dark hadrons
→ p_T^{miss} aligned with jets

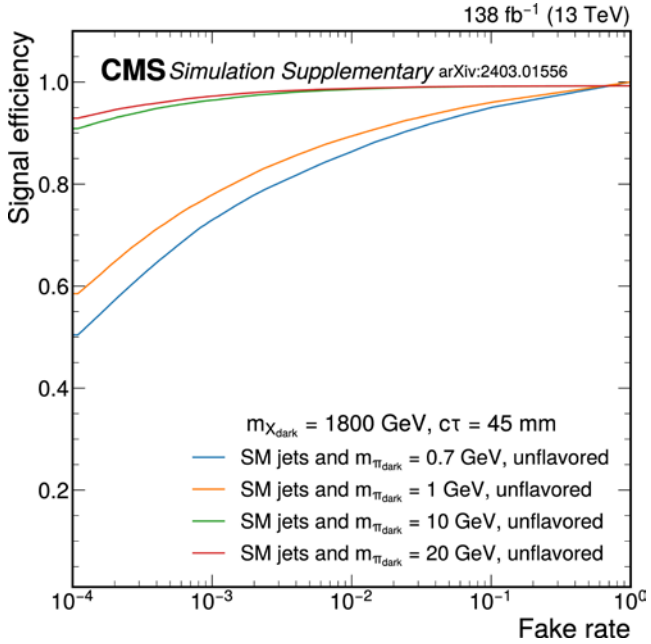
Identify with boosted decision tree trained on jet substructure variables: large improvement in sensitivity, except for very low or high dark hadron masses, where sensitivity actually *reduced*



Emerging jets (EMJs)
dark hadrons decay after some lifetime
→ multiple displaced vertices within each jet

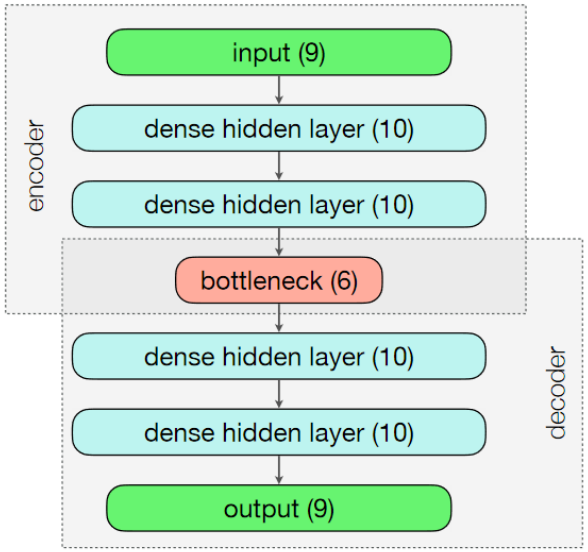


Identify with graph neural network trained on track momentum and displacement: excellent performance until dark hadron mass ≤ 1 GeV

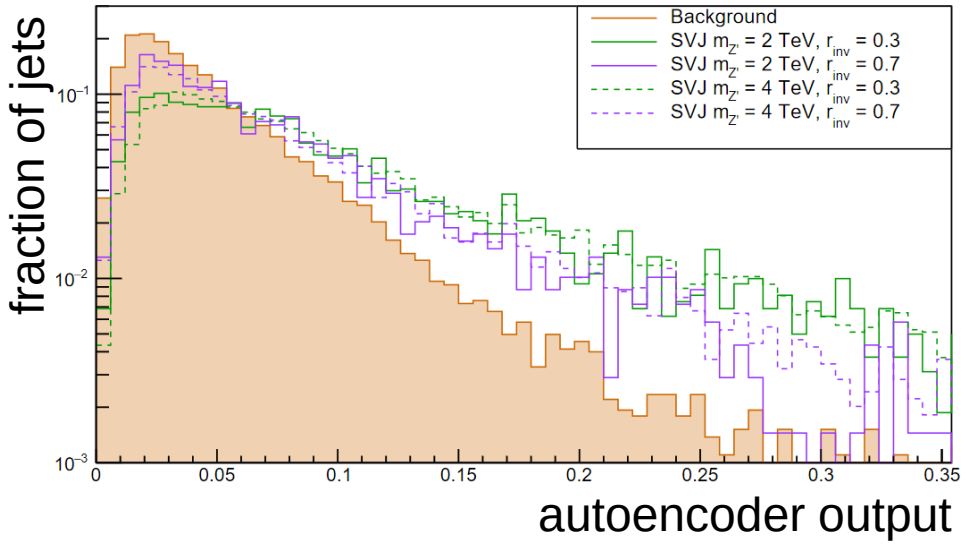

138 fb^{-1} (13 TeV)
CMS Simulation Supplementary arXiv:2403.01556
Signal efficiency
Fake rate
$m_{X_{dark}}$ = 1800 GeV, cτ = 45 mm
SM jets and $m_{\pi_{dark}}$ = 0.7 GeV, unflavored
SM jets and $m_{\pi_{dark}}$ = 1 GeV, unflavored
SM jets and $m_{\pi_{dark}}$ = 10 GeV, unflavored
SM jets and $m_{\pi_{dark}}$ = 20 GeV, unflavored

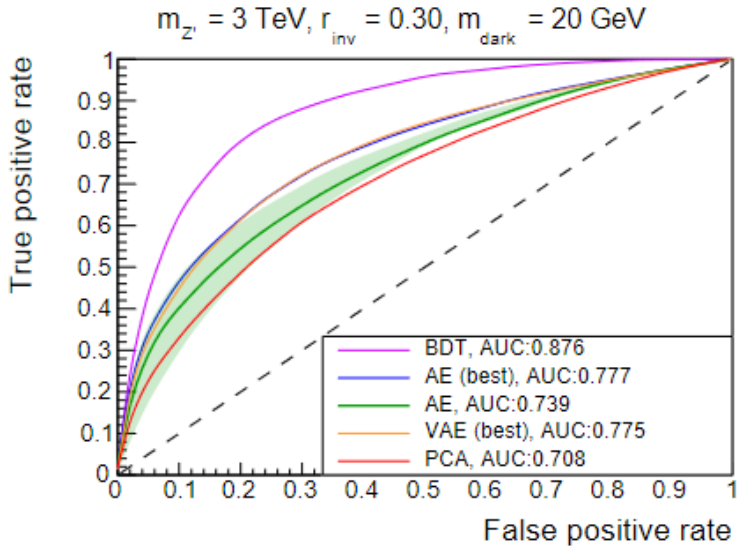
An Autoencoder for Semivisible Jets

- Create a latent representation that can be used to accurately reconstruct *background*
  - Autoencoder (AE) trained on QCD background using jet substructure variables
  - Signal not used in training; identified during inference as having high *reconstruction error*
- Compare to BDT trained on signals w/ m_{dark} = 20 GeV
  - Supervised algorithm typically performs better, but AE still improves on simpler methods like PCA



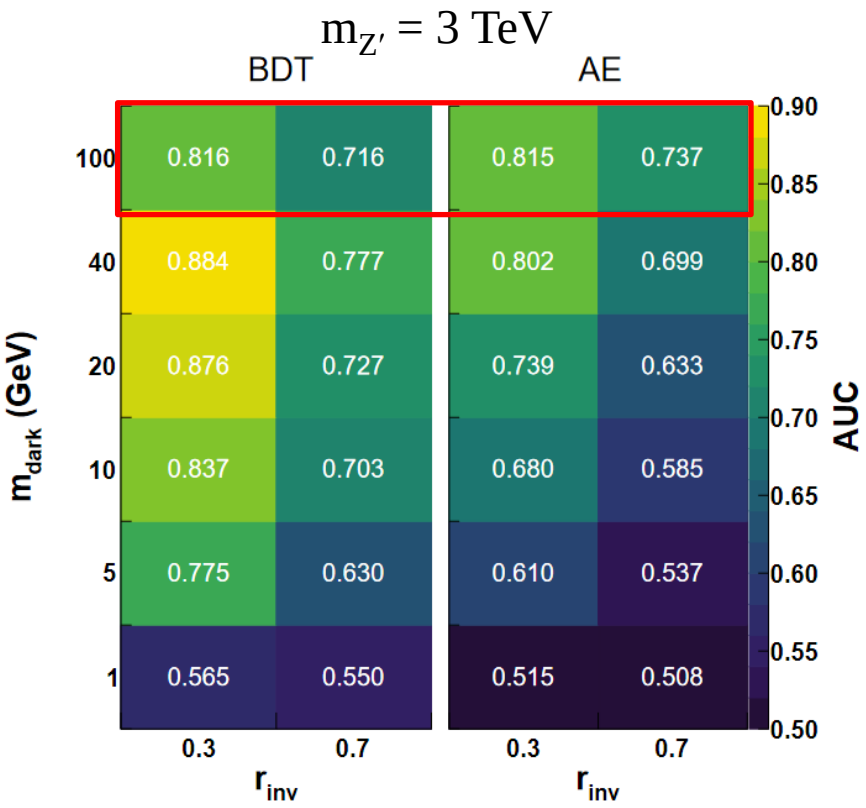
arXiv:2112.02864


Background
SVJ $m_{Z'}$ = 2 TeV, r_{inv} = 0.3
SVJ $m_{Z'}$ = 2 TeV, r_{inv} = 0.7
SVJ $m_{Z'}$ = 4 TeV, r_{inv} = 0.3
SVJ $m_{Z'}$ = 4 TeV, r_{inv} = 0.7
fraction of jets
autoencoder output

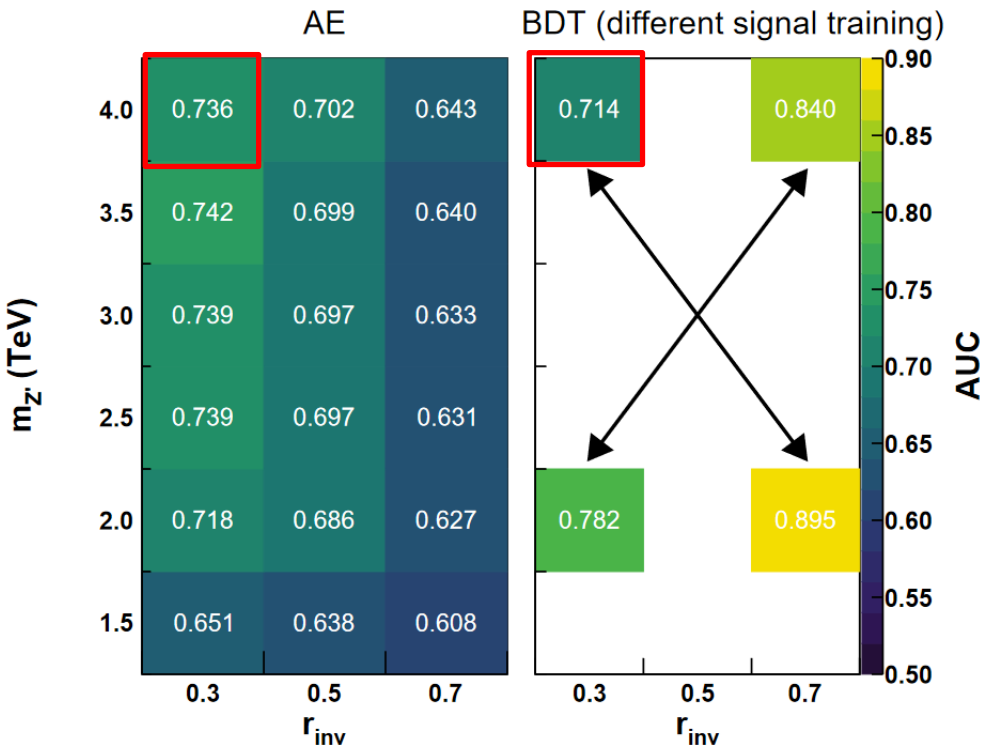


SVJ Autoencoder Performance

- Autoencoder can *outperform* BDT on signals with different m_{dark} values
  - Direct demonstration of model dependence in supervised algorithms

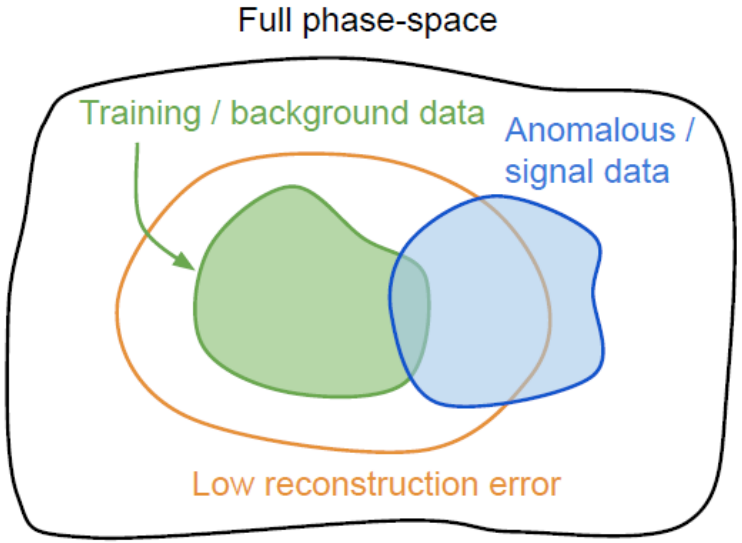


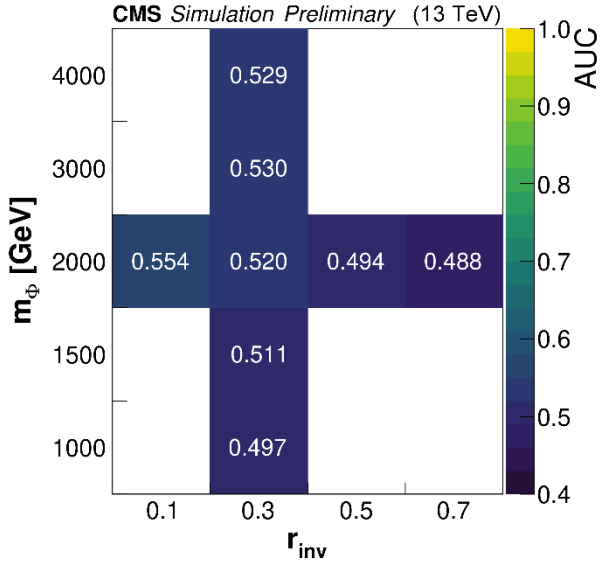
- Similarly, AE can outperform for some r_{inv} values
  - Less info for supervised training at high r_{inv}



Wasserstein Normalized Autoencoder

- Autoencoders have *complexity bias*: may learn to reconstruct any event below some complexity level
- SVJ case: easy to discriminate against QCD (lower complexity), but not against $t\bar{t}$ (higher complexity)
  - Boosted top quark decaying to $\ell\nu b$ is closest SM analogue to SVJ with real p_T^{miss}

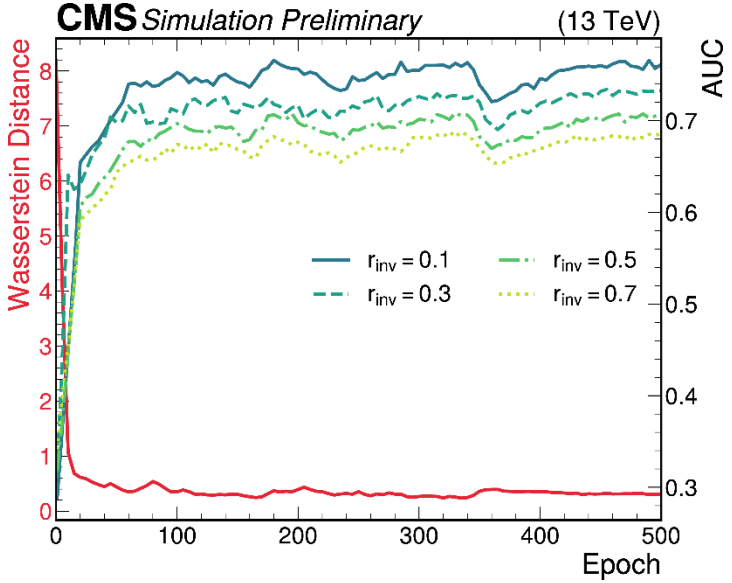


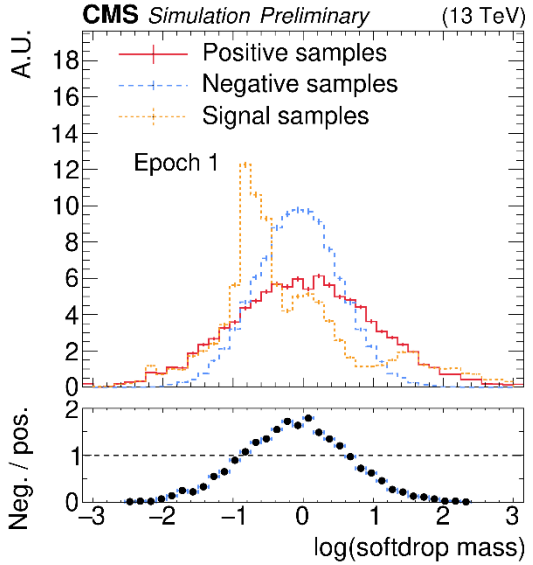


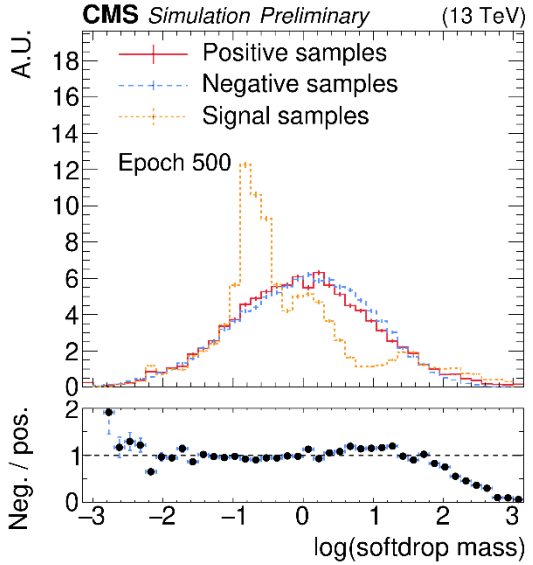
- Use *energy-based model* formalism: define p_θ so regions with *low reco error* E_θ have *high probability*
  - Learn $p_\theta = p_{data}$ by sampling from p_θ and comparing to data samples → *normalized autoencoder*
    - Comparisons like $E_{data} - E_\theta$ are unstable: can diverge or collapse

➢ *Directly minimize* Wasserstein distance (Earth Mover's Distance) between data and θ distributions

WNAE Performance

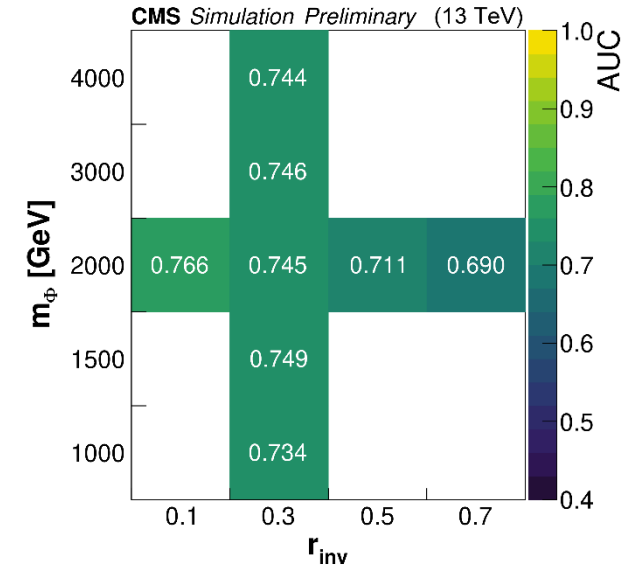






- Low Wasserstein distance corresponds directly to high AUC for signals
  - But Wasserstein distance only computed on background!
- p_θ (negative) approaches p_{data} (positive) as training converges

➢ Final result: equally strong performance against $t\bar{t}$!



Trigger-Level Autoencoders

- Existing trigger strategies use basic kinematic quantities: jet p_T, H_T, p_T^{miss}
  - Can be model- or production mode-dependent: influenced by mediator mass, r_{inv}, etc.
  - Thresholds have to be increased to handle higher event rates and pileup
- CMS has deployed two anomaly detection algorithms in L1 trigger during Run 3:
  - AXOL1TL (Anomaly eXtraction Online Level-1 Trigger Lightweight)
    - Variational autoencoder trained on all global L1 bits

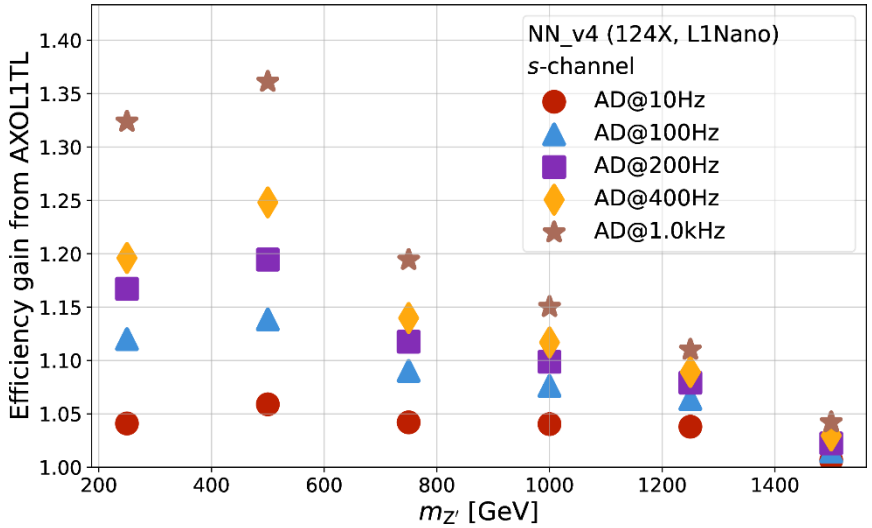
  - CICADA (Calorimeter Image Convolutional Anomaly Detection Algorithm)
    - Convolutional autoencoder trained on calorimeter energy deposits

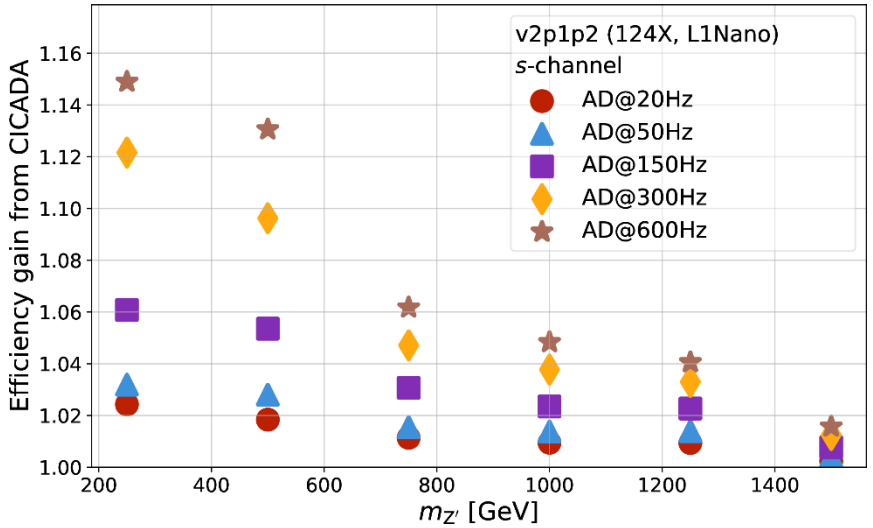
  - ➢ Trained on zero bias data to learn distribution of “uninteresting” events
  - Deployed in hardware trigger using hls4ml

Anomaly Trigger Performance

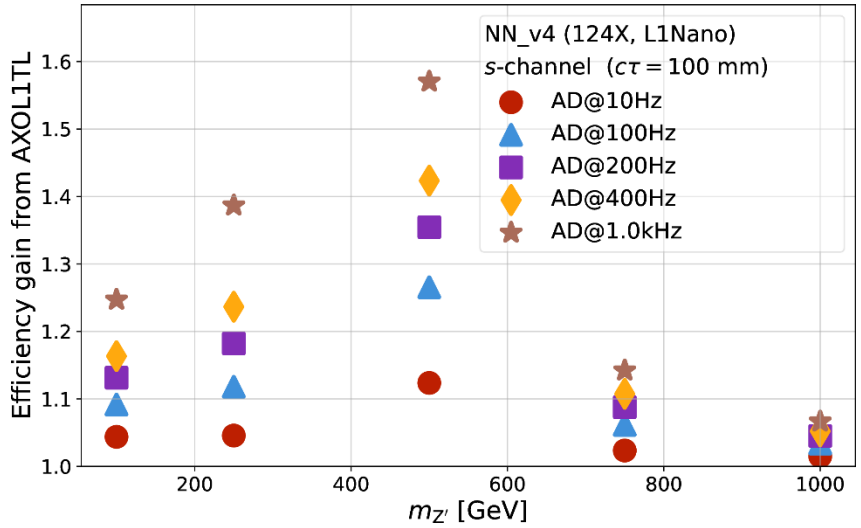
AXOL1TL | **CICADA**

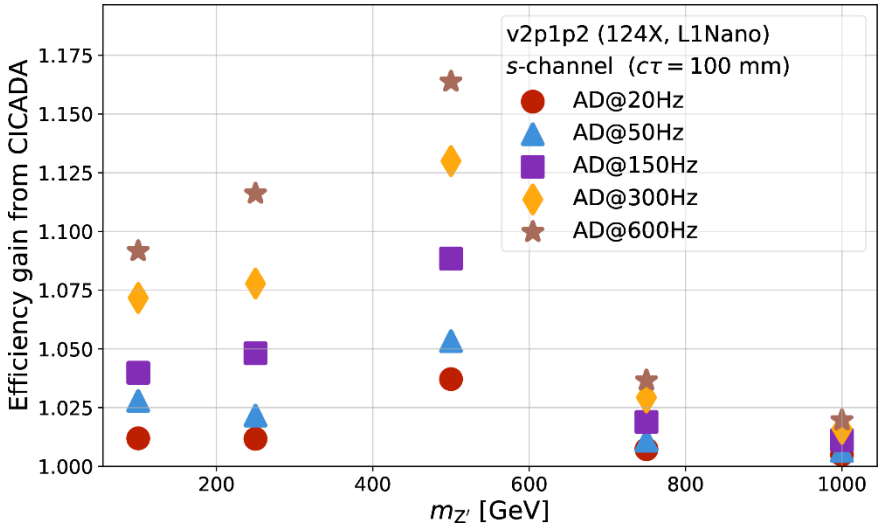
SVJ





EMJ

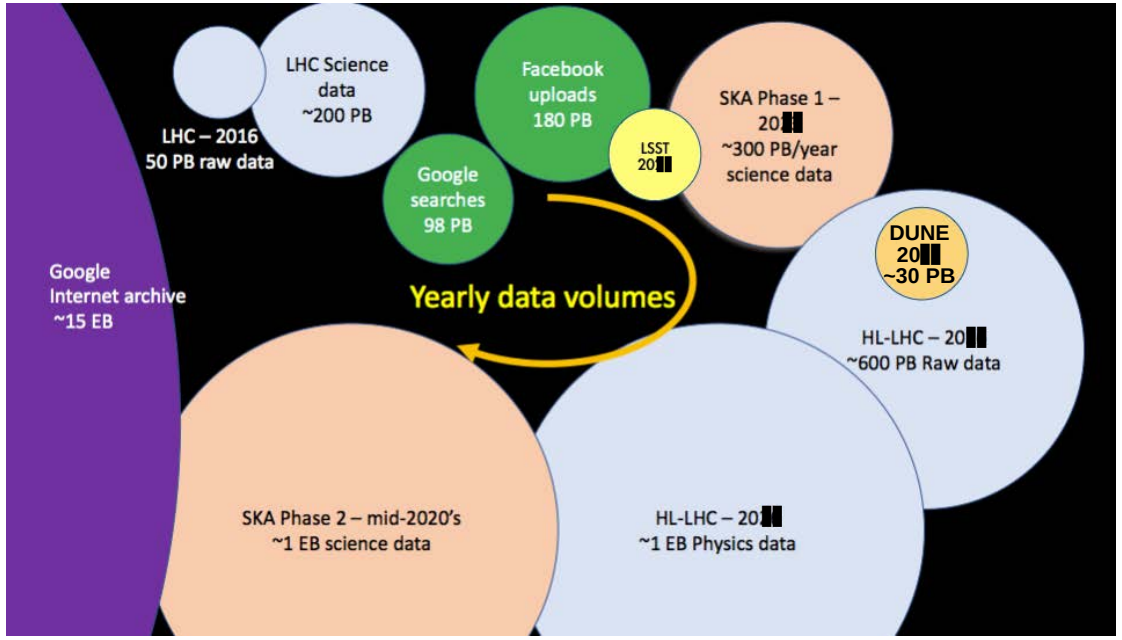




- Anomaly triggers provide significant efficiency gains (when combined w/ standard jet triggers)
- Enables access to low-mass Z' mediators
- Gains ultimately depend on allowed rate – hope to increase!
- New versions being trained to improve background rejection
  - Including exploration of WNAE approach

Generative Learning

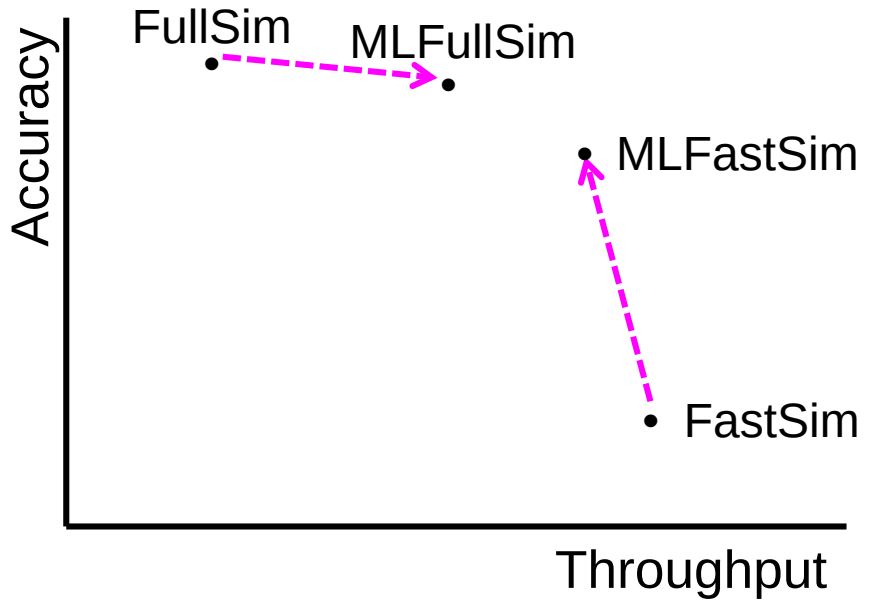
ML for Simulation


LHC Science data ~200 PB
LHC – 2016 50 PB raw data
Facebook uploads 180 PB
Google searches 98 PB
LSST 20
SKA Phase 1 – 20 ~300 PB/year science data
DUNE 20 ~30 PB
Google Internet archive ~15 EB
Yearly data volumes
HL-LHC – 20 ~600 PB Raw data
SKA Phase 2 – mid-2020's ~1 EB science data
HL-LHC – 20 ~1 EB Physics data

- Upcoming experiments (High Luminosity LHC, DUNE, LSST, Square Kilometer Array) will deliver 10× or more data vs. existing experiments
- Simulation is computationally intensive & costly
  - Match growing data volumes & improved detectors: simulate *more events* with *more complexity* and *higher accuracy*
  - Computing budgets increasingly constrained

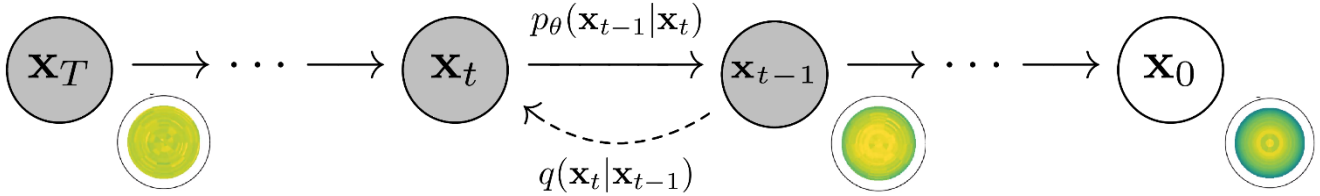
- Two classes of simulation engines:
  - FullSim: model ~every interaction precisely
  - FastSim: parametrize interactions

➢ Replace components with generative ML
  - FullSim: increase throughput, preserve accuracy
  - FastSim: increase accuracy, preserve throughput



Diffusion Models

1. Add known amount of random noise to input training data
2. Learn to predict noise in training data
   - Either directly, or via “score function” (gradient of probability density)
     - Equivalent for variance-preserving score formulation
3. Starting from pure random noise, remove predicted noise iteratively
   → create a new sample from distribution of training data



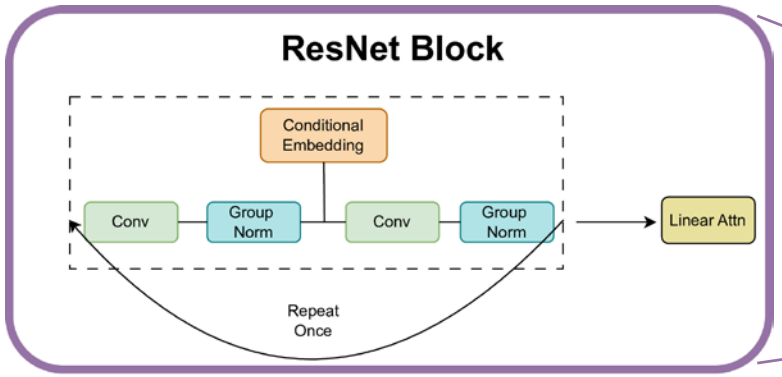
- Rapidly adopted for image generation in industry
  - High-quality output, stable training
- Particle showers in calorimeters – equivalent problem?
  - Try to shortcut both computationally expensive electromagnetic physics interactions and geometry navigation

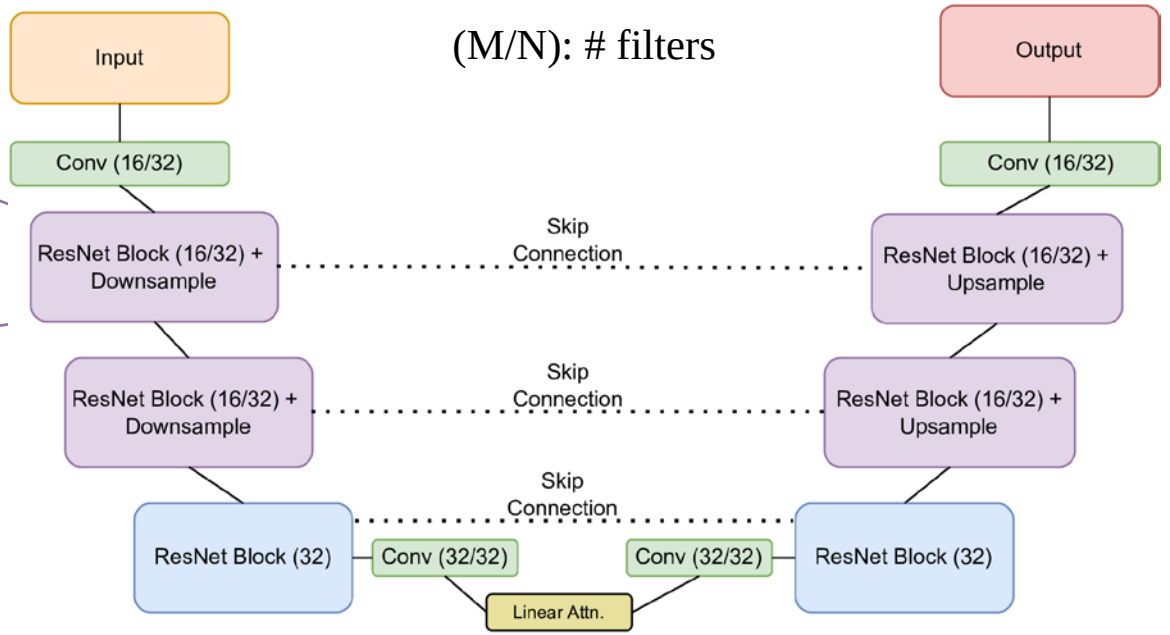
DreamStudio

DALL·E 3

Midjourney

CaloDiffusion



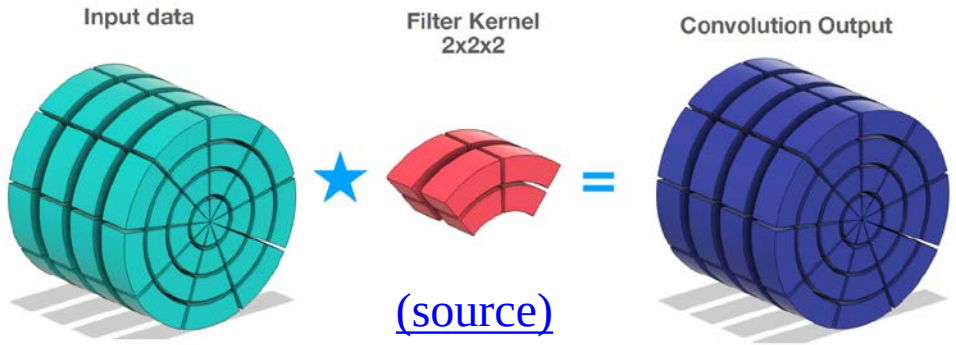


- Base architecture: U-net
  - Skip connections ensure no loss of information
- Linear self-attention layers applied to each convolutional ResNet block
  - Allows dimensionality reduction in *z* to handle longitudinal correlations in showers
- + several geometric innovations (next slides)

- Objective:
  - Predict weighted average of noise and denoised image (energy per pixel)
- Aim for highest achievable quality first
  - Then focus on improving speed
  - Wrong answers can be obtained infinitely fast

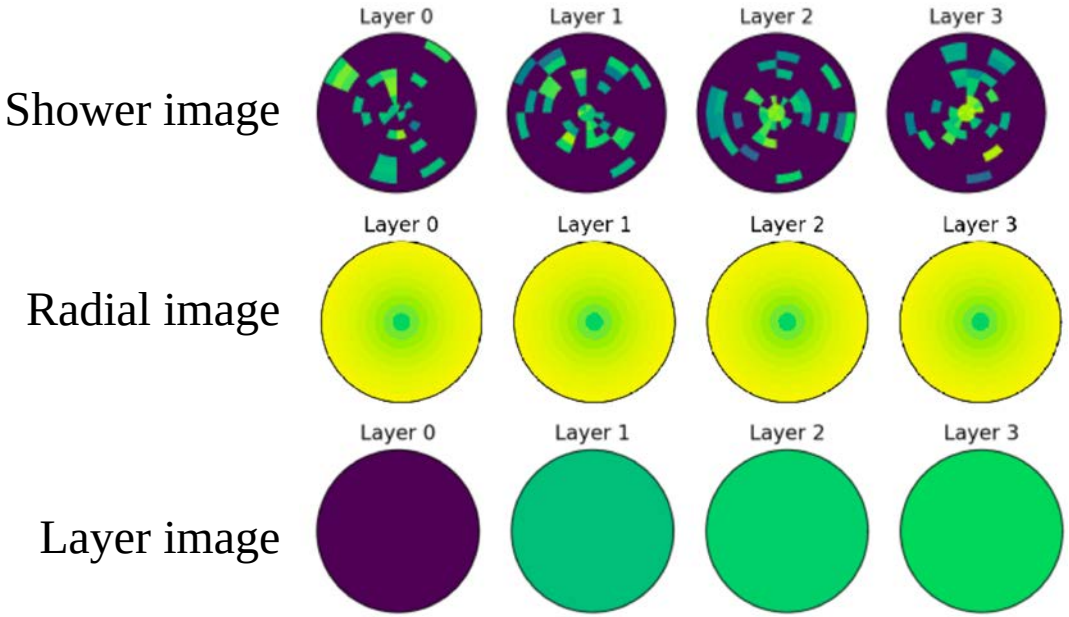
Geometric Innovations

- Particle showers are invariant & periodic in φ
  - Pad in φ so convolutions "wrap around"



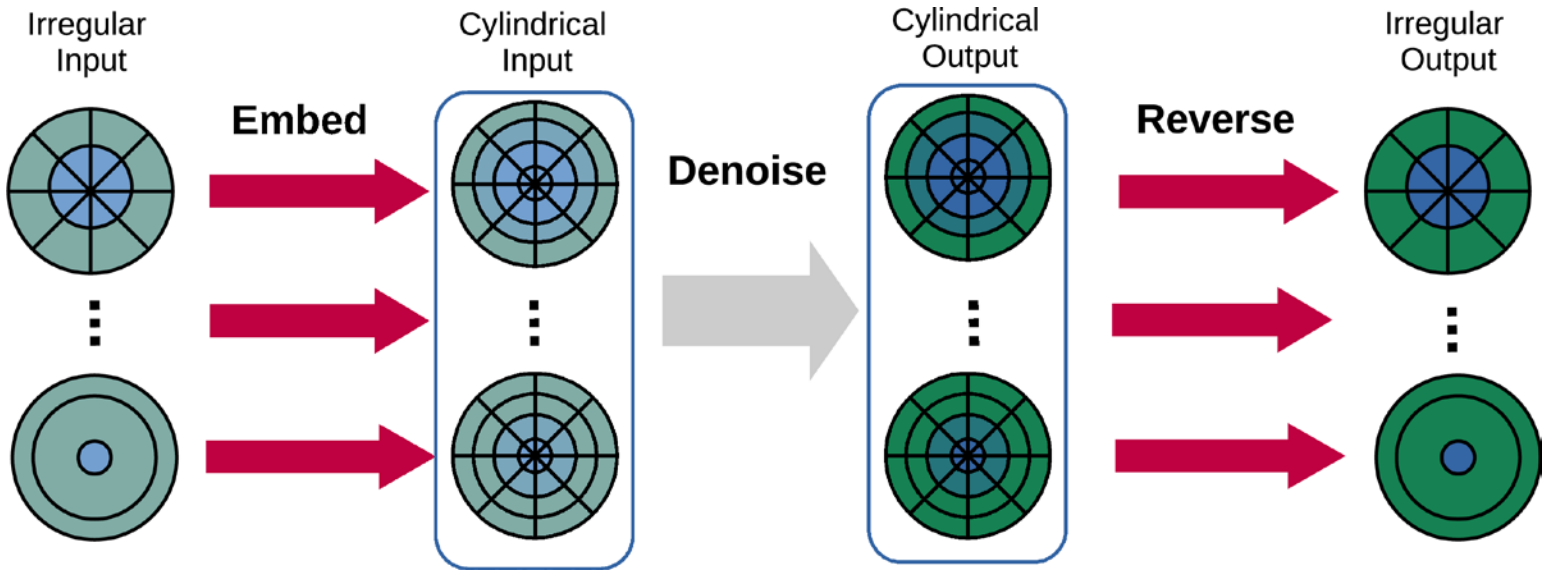
(source)

- Particle showers are *not* invariant in *r* or *z*
  - Provide *r* and *z* (layer) as extra per-pixel channels (input features)
  - Convolutions become *conditional*



➢ *Conditional cylindrical convolutions*
  - Handle inherent features of particle detector geometry, distinct from rectangular images

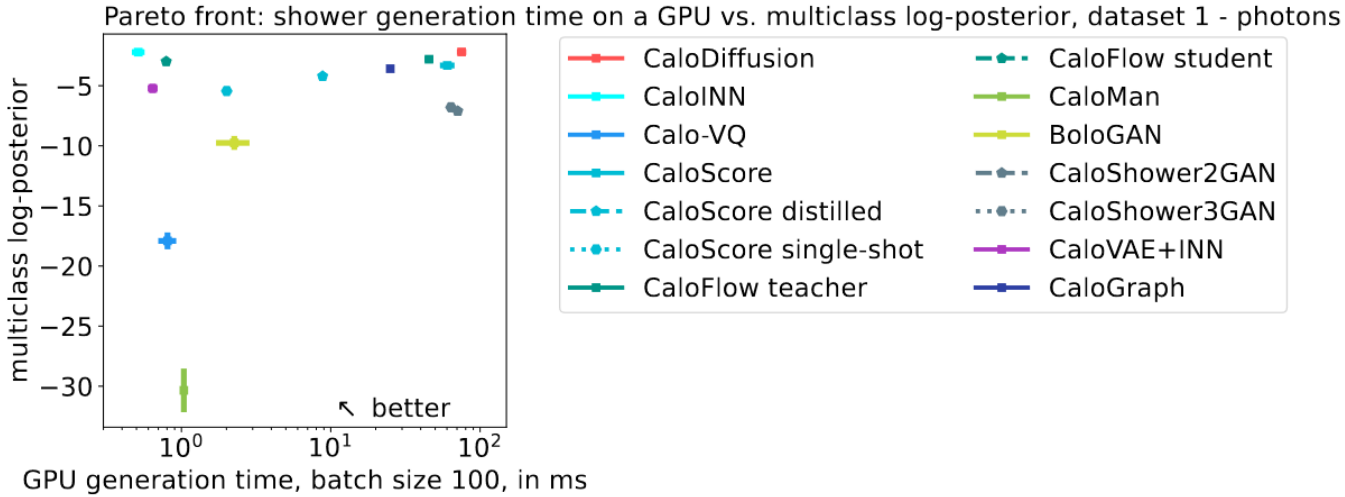
Geometry Latent Mapping: GLaM

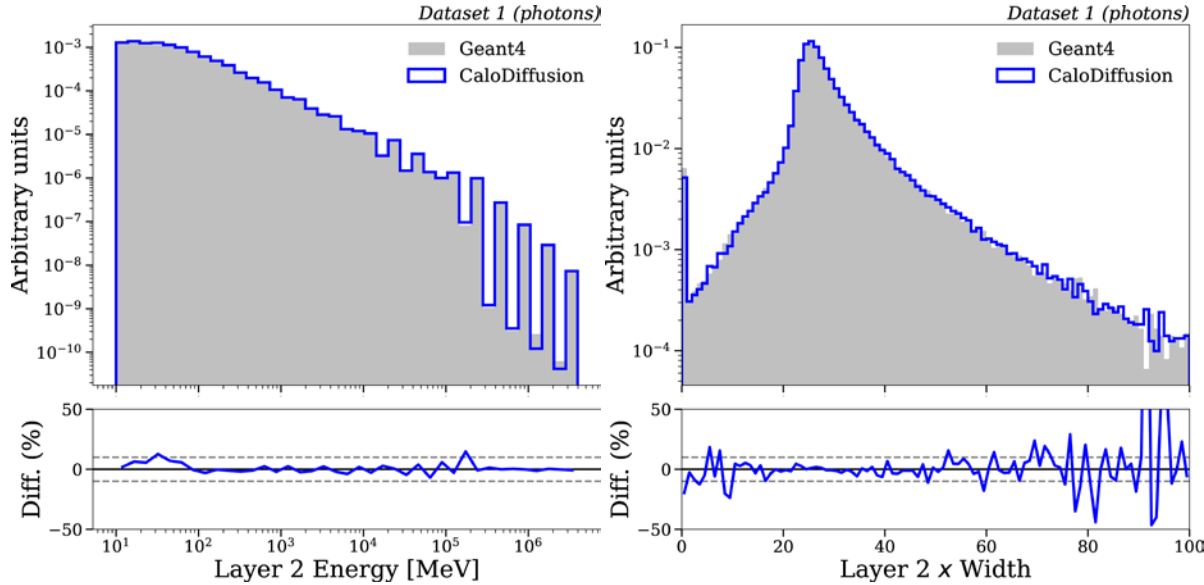


- Some calorimeter geometries have different radial/angular bins in each layer
  - Can't directly apply convolutions, which require regular neighbor structure
- Learn forward and reverse embeddings to and from a regular geometry
  - Simple matrices C (NxM) and D (MxN)
    - C initialized to split or merge cells based on overlap between original and embedded geometries
    - D initialized as Moore-Penrose pseudoinverse of C
- Inspired by "latent diffusion" approach
  - But not necessarily lower-dimensional representation; can be higher-dimensional

CaloChallenge Performance

- CaloChallenge: community competition w/ three public datasets (~200K events each):
  1. Low granularity (368/533 voxels), irregular geometry (based on ATLAS calorimeter), photon & pion showers
  2. Medium granularity (6480 voxels), silicon-tungsten sampling calorimeter, electron showers
  3. High granularity (40500 voxels), otherwise same as #2
- CaloDiffusion: leading performance in accuracy
  - Slower inference time, typical of diffusion models (noise removed iteratively, multiple steps required)
  - The first version required 400 diffusion steps



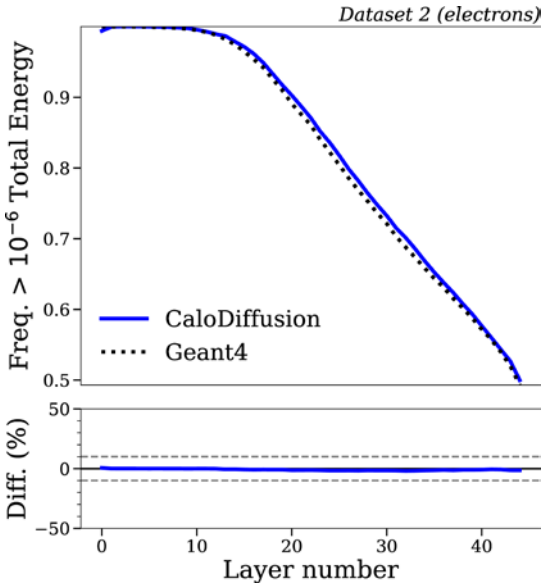


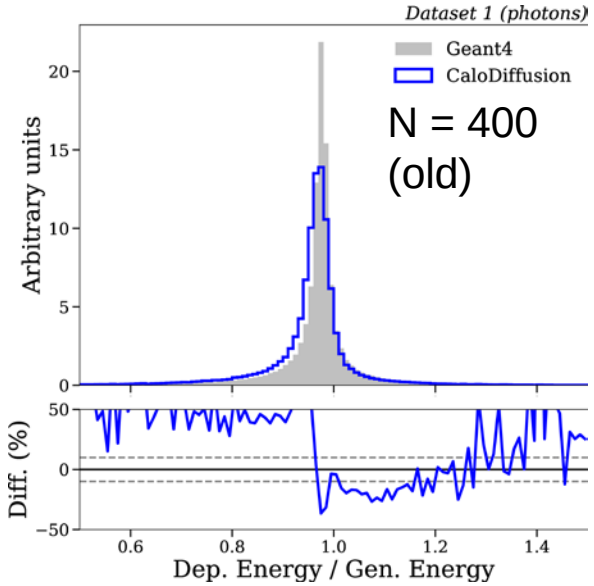
Improvement: More (& Better) Diffusion!

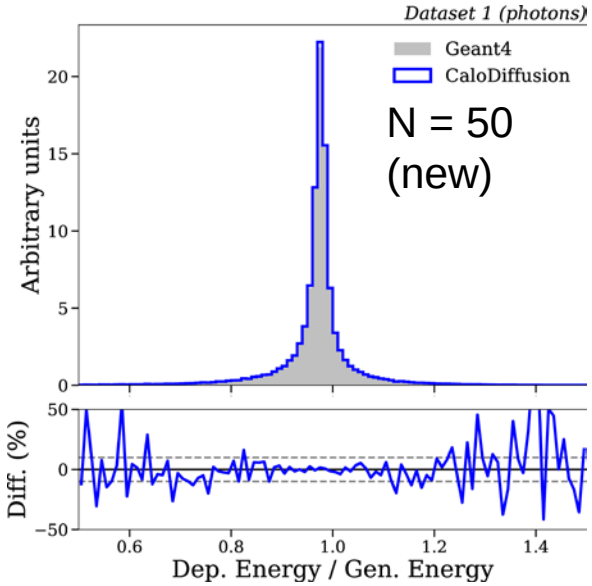
- Train LayerDiffusion to predict energy deposited per layer (1D diffusion)
  - Negligible inference time (200 steps) compared to CaloDiffusion
- Normalize CaloDiffusion output based on LayerDiffusion
  - Only if both models predict sufficiently non-zero deposited energy in a layer

➢ Substantial improvement in total energy modeling

- Number of CaloDiffusion steps can be reduced 4–8× with no loss of quality
  - Directly translates to 4–8× speedup



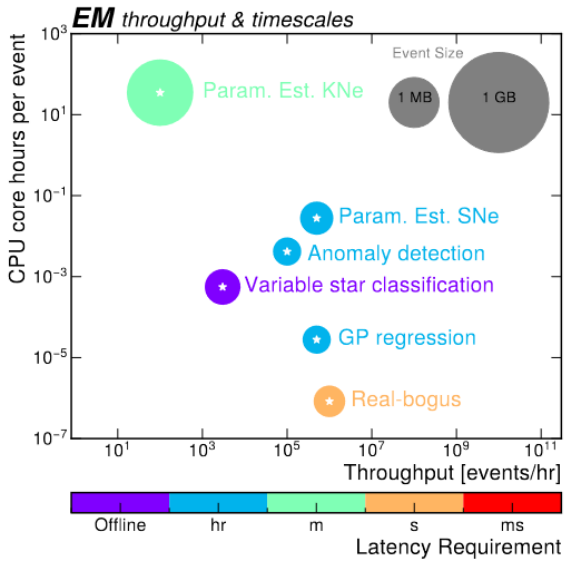


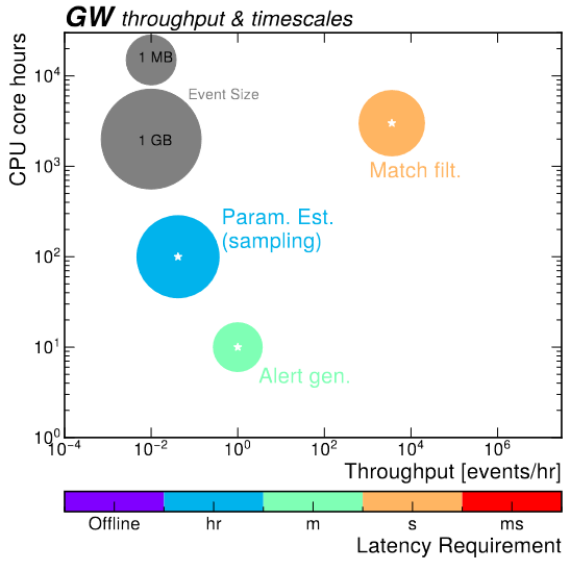


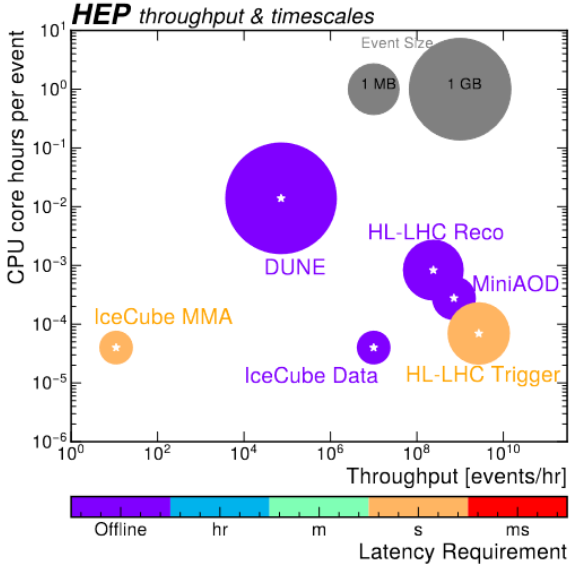
- Further improvements in training and inference:
  - Minimum signal to noise ratio weighting during training (arXiv:2401.13162)
  - Improved noise schedule ("EDM", arXiv:2206.00364)
  - Deterministic sampling algorithm ("DDIM", arXiv:2010.02502)

Inference as a Service

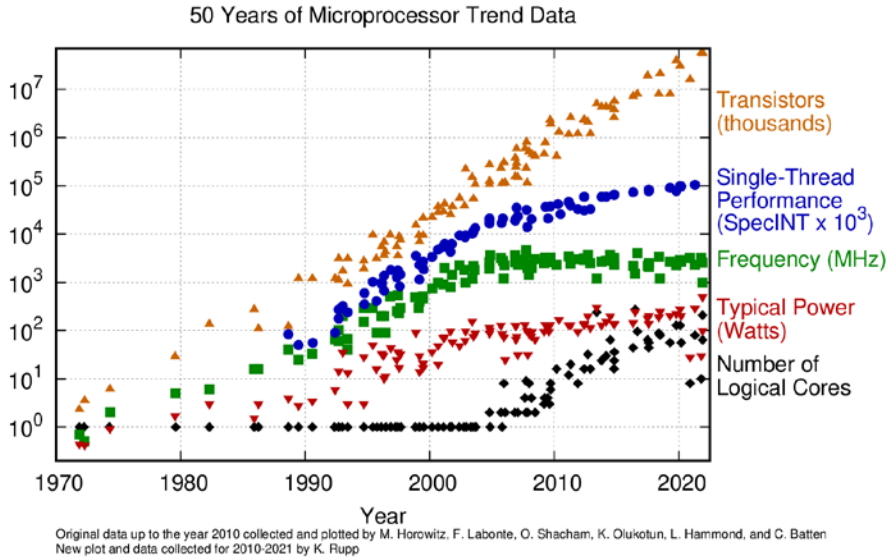
Computing for HEP & MMA



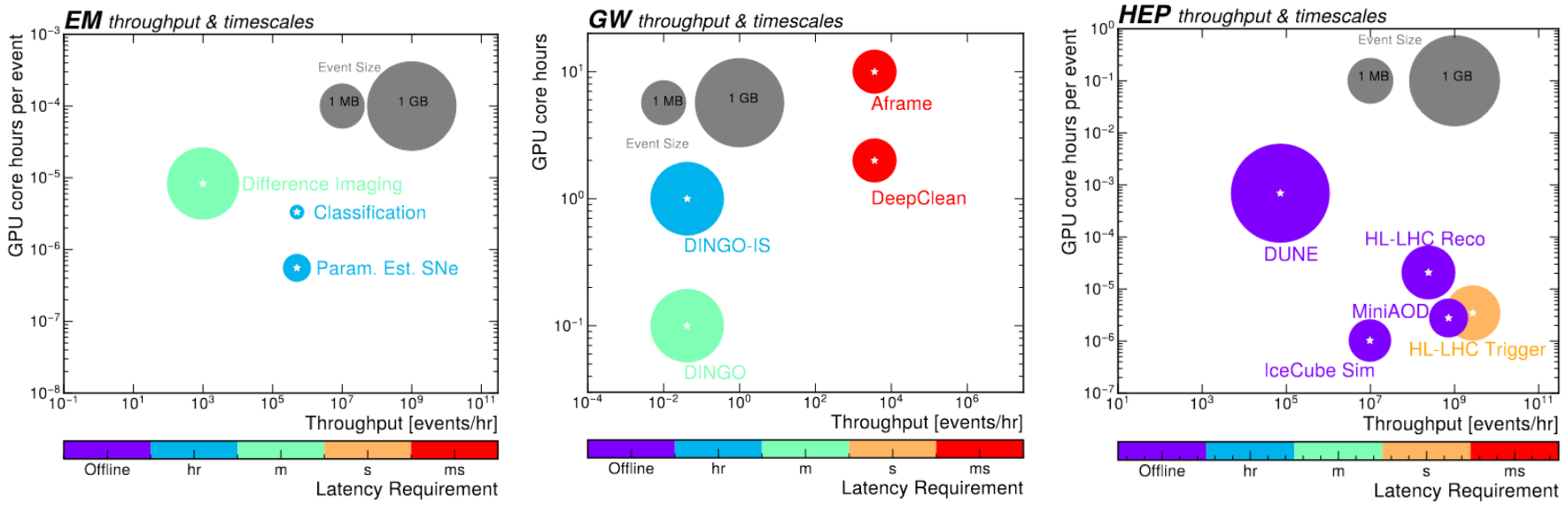




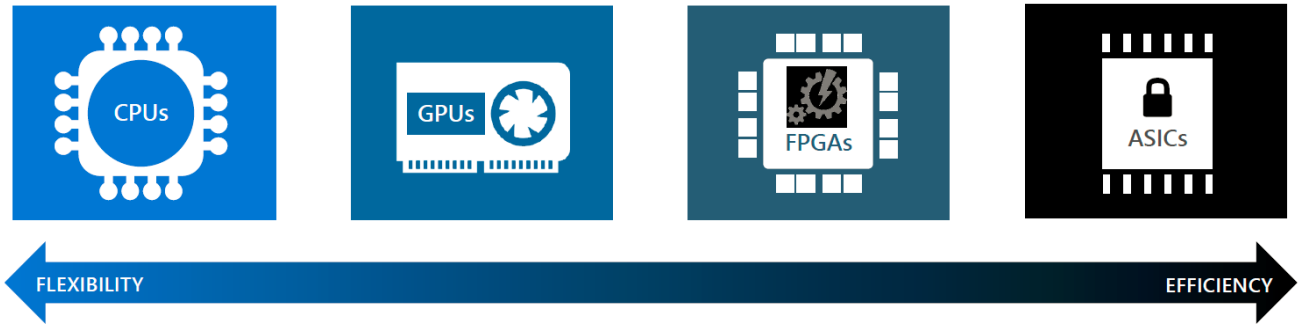
- Electromagnetic astronomy, gravitational-wave astronomy, and high-energy physics data processing:
  - All involve numerous tasks (classical or ML)
  - Various latency & throughput requirements, dataset sizes
- CPU **stagnation**: Moore's Law continues, but Dennard scaling has broken down
  → performance/thread can't keep up with demand


50 Years of Microprocessor Trend Data
Transistors (thousands)
Single-Thread Performance (SpecINT x 10^3)
Frequency (MHz)
Typical Power (Watts)
Number of Logical Cores
Year
Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2021 by K. Rupp

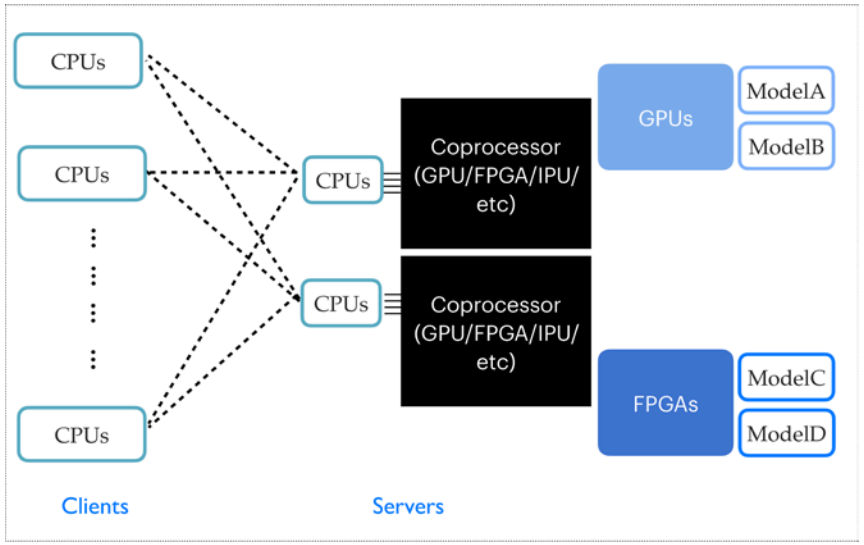
Heterogeneous Computing



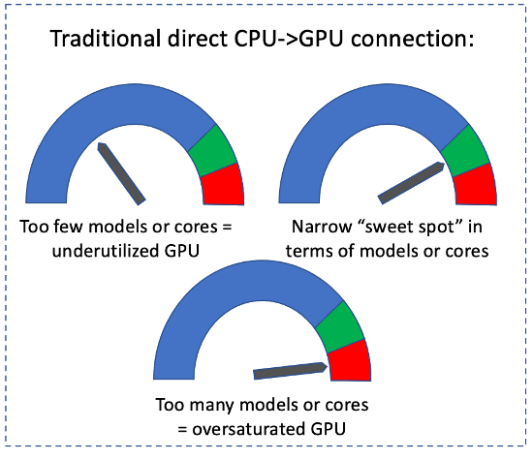
- Heterogeneous **revolution**: rise of *specialized coprocessors* attached to general-purpose CPUs
  - **GPUs**: reduce core hour usage, improve throughput
  - **FPGAs**, **ASICs**: low-latency applications (trigger, on-detector readout)
- *Deep learning* uses limited set of mathematical operations: perfect for acceleration on GPUs etc.
  - *Inference as a service*: most general/abstract way to offload tasks to coprocessors

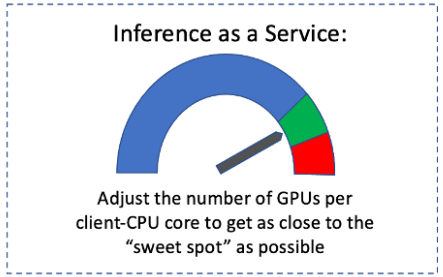


Services for Optimized Network Inference on Coprocessors



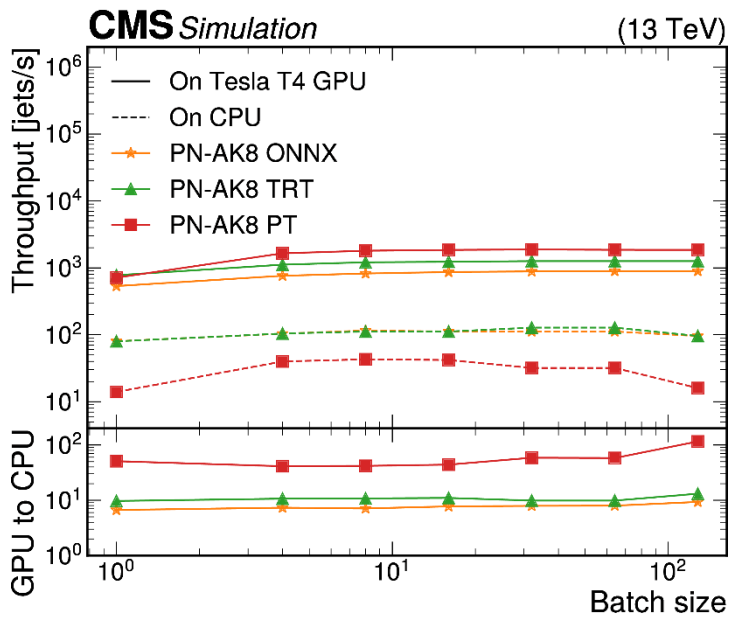
- **SONIC**: *design pattern* to implement inference as a service in experiment software frameworks
  - Build on industry technologies: gRPC communication, Nvidia Triton inference server
- Advantages:
  - *Isolation*: factorize ML frameworks out of experiment software
  - *Simplicity*: client code only handles input/output conversions
  - *Flexibility*: CPU-GPU ratios can be adjusted dynamically
  - *Efficiency*: optimize CPU-GPU ratios to ensure full usage
    - *Dynamic batching* especially useful for GPUs
  - *Portability*: use CPU, GPU, FPGA, etc. with no code changes
  - *Accessibility*: use remote coprocessors (if none available locally)

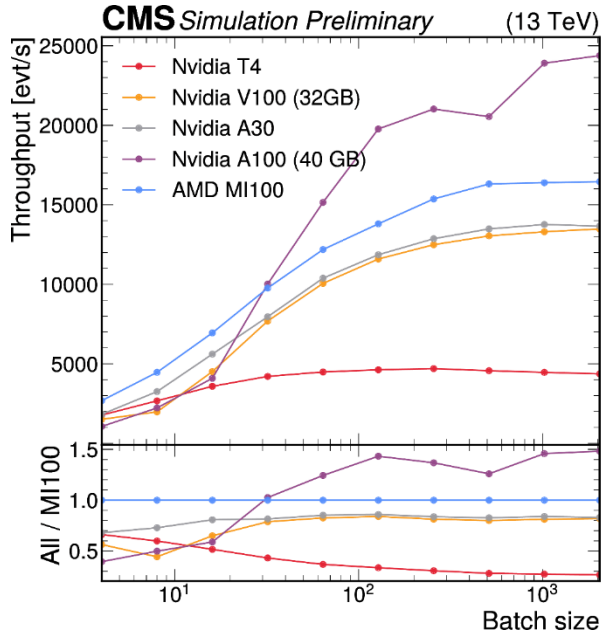

Traditional direct CPU->GPU connection:
Too few models or cores = underutilized GPU
Narrow "sweet spot" in terms of models or cores
Too many models or cores = oversaturated GPU

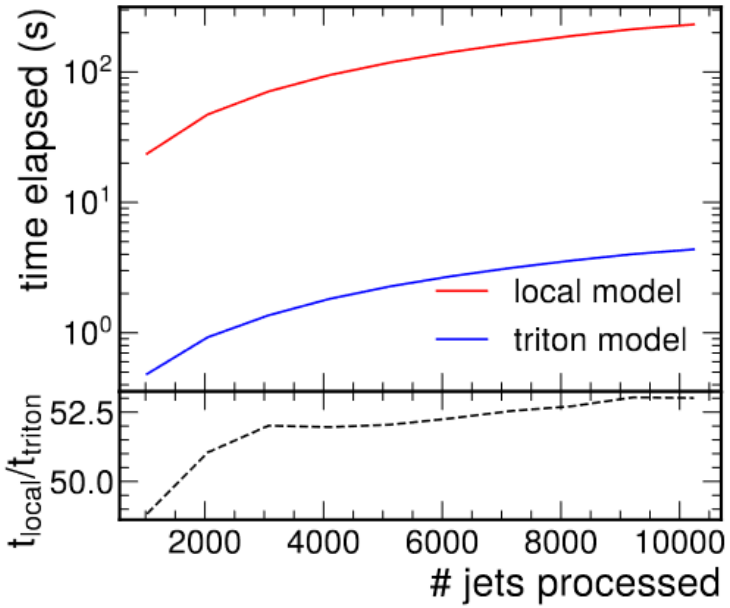


SONIC for Classification

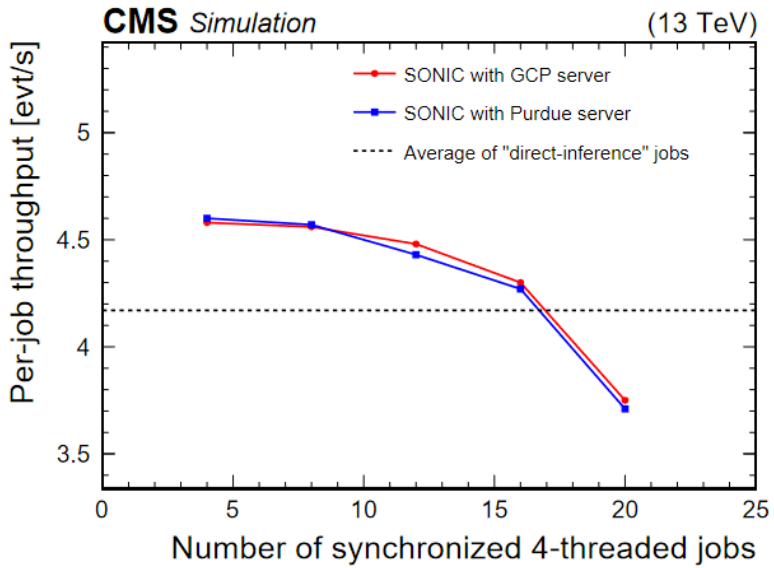
- ParticleNet: graph neural network for jet classification
  - Measure Higgs boson properties, search for new physics (including emerging jets!)
- In central CMS data processing workflows: **10–50× speedup** on GPU
- In offline user analysis workflows: **50× speedup** maintained
- Port to AMD GPU: ~2× improvement vs. most Nvidia GPUs currently in use
  - Other algorithms have been ported to FPGA, TPU, IPU using SONIC – with *no code changes*





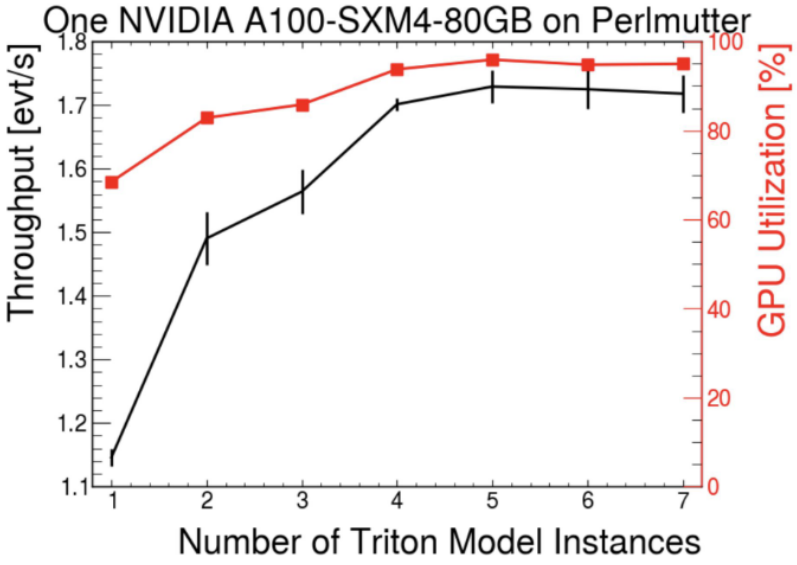


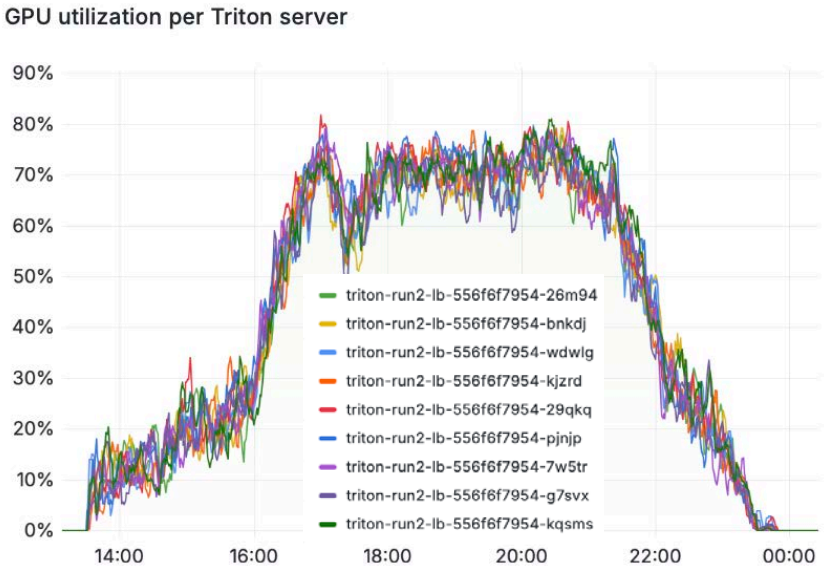
SONIC at Scale



Server location irrelevant (at least up to 100s of km, Indiana → Iowa): latency from transmission hidden by asynchronous non-blocking calls

High GPU utilization achieved for Exa.TrkX: (large) graph neural network for charged track reconstruction (on CPU: 0.1 evt/s)





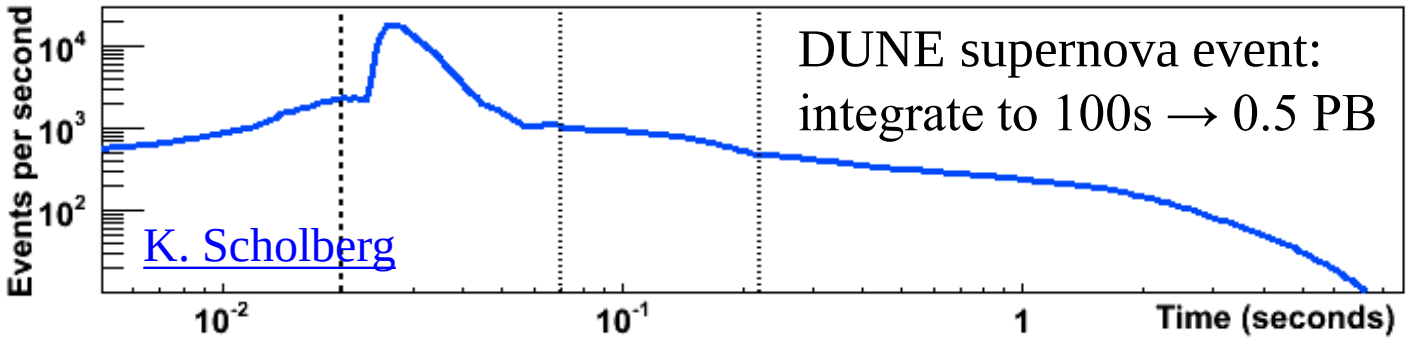
Consistent and uniform load across GPUs for hours, in more production-like scenario w/ grid jobs arriving over time

SuperSONIC

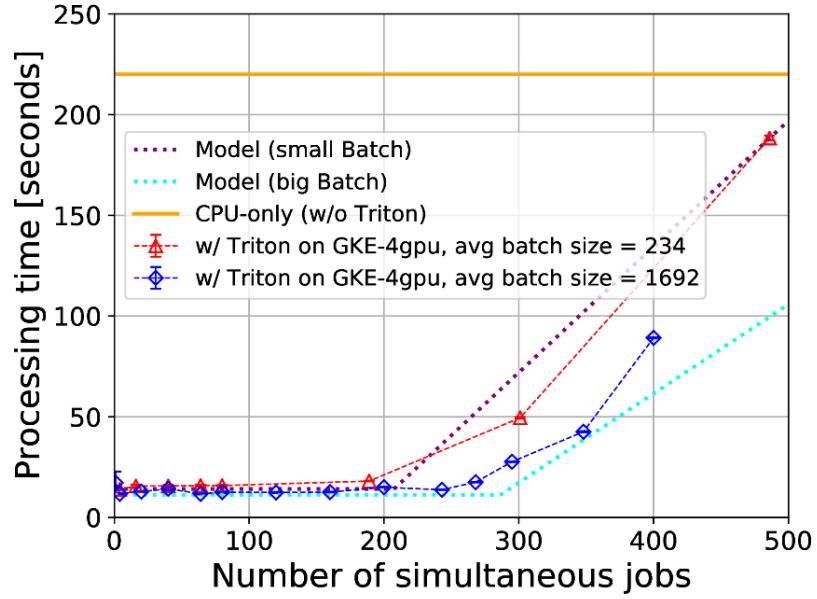
Shared Helm chart w/ Kubernetes configuration, load balancing, authentication, etc.

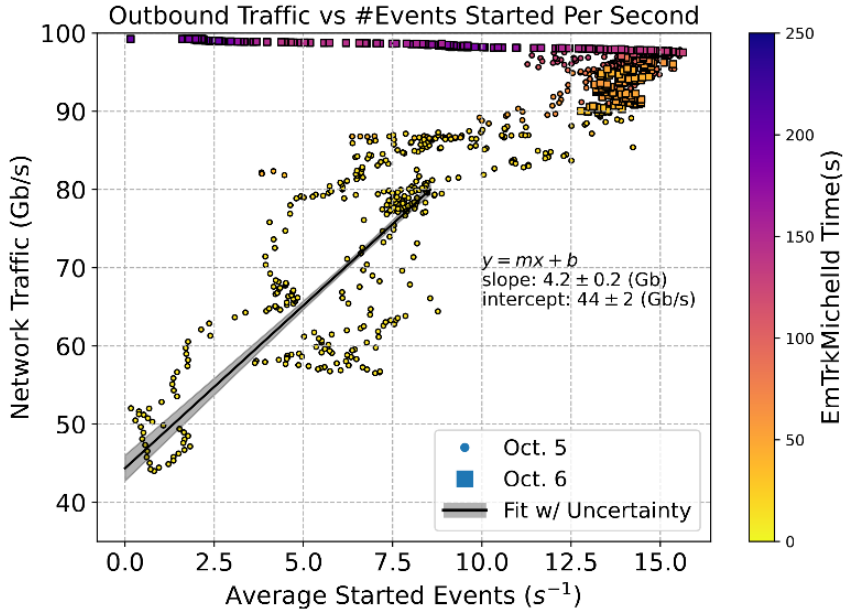
SONIC for Neutrinos

- ProtoDUNE: small-scale version of DUNE, at CERN (800 tons LAr)
- Reconstruction dominated by large convolutional neural network
  - Classifies hits as track, shower, or Michel electron
- Algorithm level: 17× speedup using GPU
  - Overall reconstruction time reduced by 2.7×
- Datacenter level: reprocess entire protoDUNE dataset in weeks instead of months
  - Mitigate network saturation (4 Gb/image) via concurrency limits
- Even more important for MMA uses:



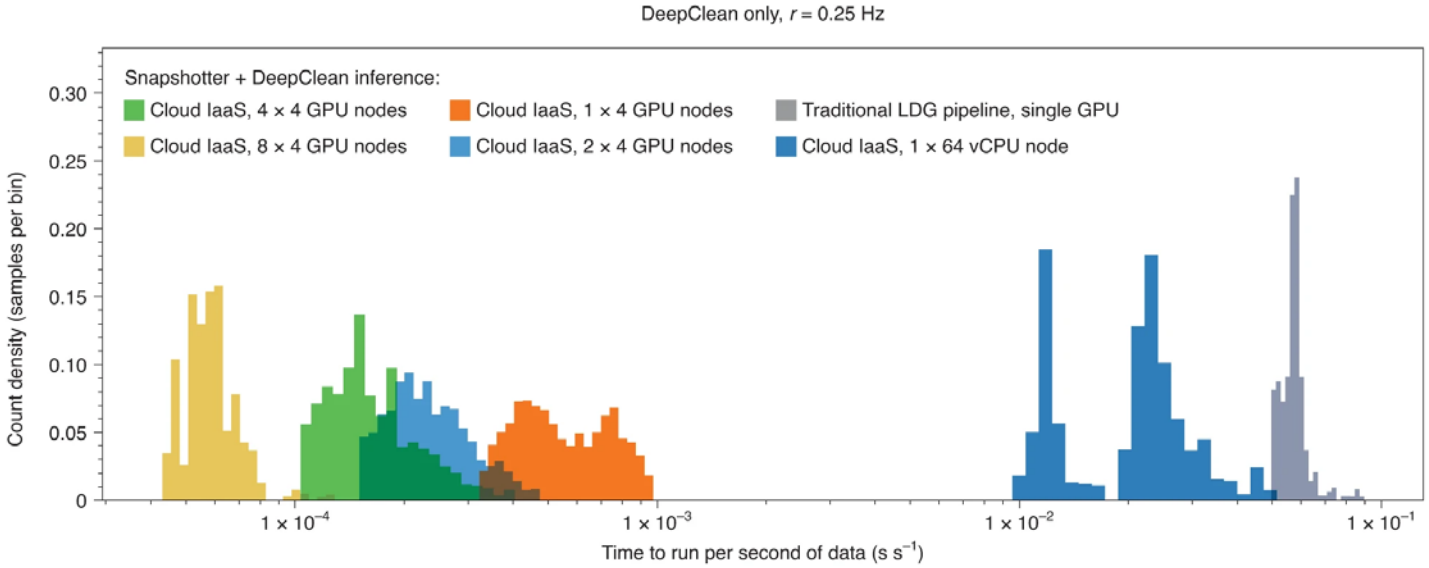
➢ Now being explored for IceCube (using SuperSONIC)

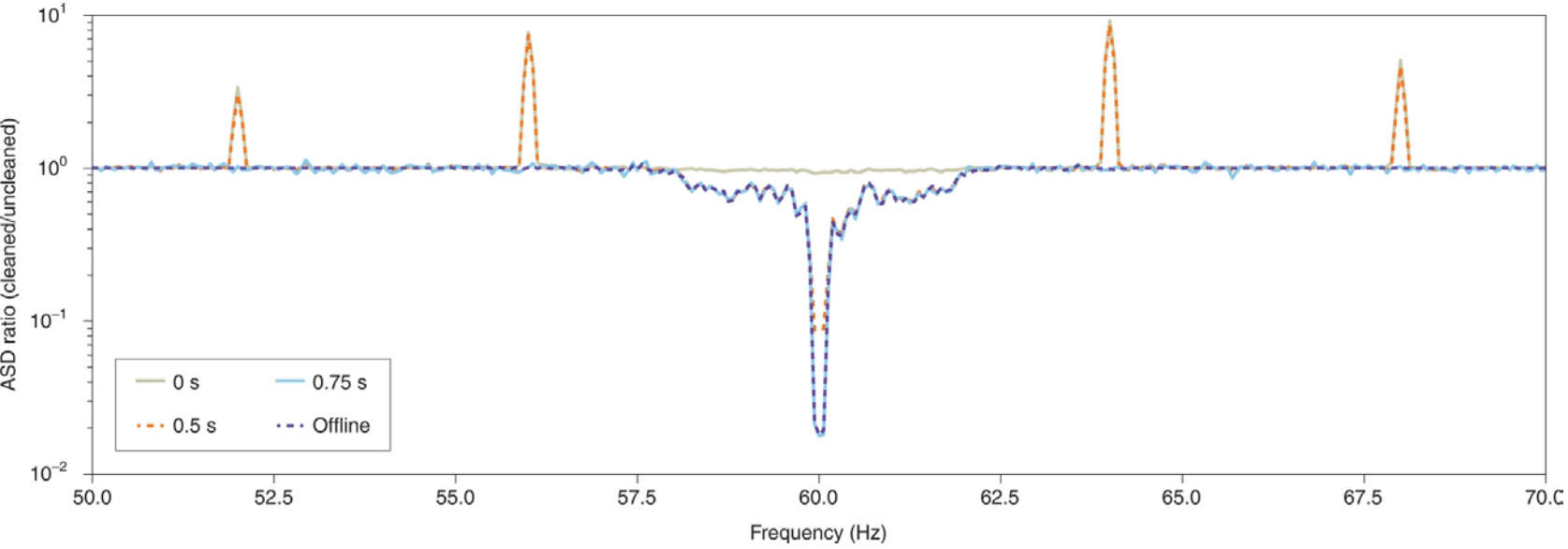

Processing time [seconds]
Number of simultaneous jobs
Model (small Batch)
Model (big Batch)
CPU-only (w/o Triton)
w/ Triton on GKE-4gpu, avg batch size = 234
w/ Triton on GKE-4gpu, avg batch size = 1692



SONIC for Gravitational Waves

- Multiple order of magnitude speedup in *online* ML-based processing of LIGO data
  - Enabled by seamless scaling into GPU resources and parallel processing of inference requests

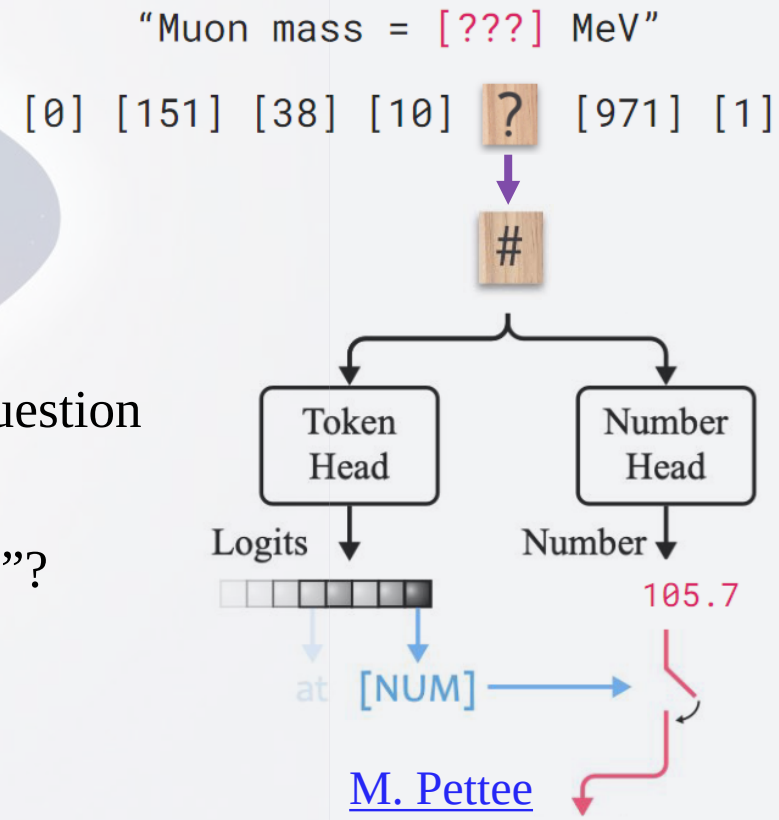

DeepClean only, r = 0.25 Hz
Snapshotter + DeepClean inference:
Cloud IaaS, 4 × 4 GPU nodes
Cloud IaaS, 1 × 4 GPU nodes
Traditional LDG pipeline, single GPU
Cloud IaaS, 8 × 4 GPU nodes
Cloud IaaS, 2 × 4 GPU nodes
Cloud IaaS, 1 × 64 vCPU node
Count density (samples per bin)
0.30
0.25
0.20
0.15
0.10
0.05
0
1 × 10⁻⁴
1 × 10⁻³
1 × 10⁻²
1 × 10⁻¹
Time to run per second of data (s s⁻¹)



- LIGO DeepClean algorithm denoises *streaming* data
  - Small latency introduced online to match offline performance

Outlook

- Learning and using *probability densities* is the new paradigm in artificial intelligence
- *Unsupervised and generative learning* let us ask and answer whole new classes of questions
  - For high energy physics and multimessenger astronomy: protons, neutrinos, photons, gravitational waves
- *Inference as a service* is the optimal way to accelerate AI algorithms on coprocessors
- What's next?
  - Deploy latest developments (e.g. WNAE, CaloDiffusion) at scale using inference as a service
  - Continue improving and expanding usage of these new approaches
  - *Foundation models*:
    - "Generative pre-training": model learns enough to answer any question
      - Via tokenizing, finetuning, low-rank adaptation…
    - Generalize "predict the next token" to "predict the missing value"?
    - Physics values are often continuous rather than discrete → special tokens for numerical embedding (arXiv:2310.02989)

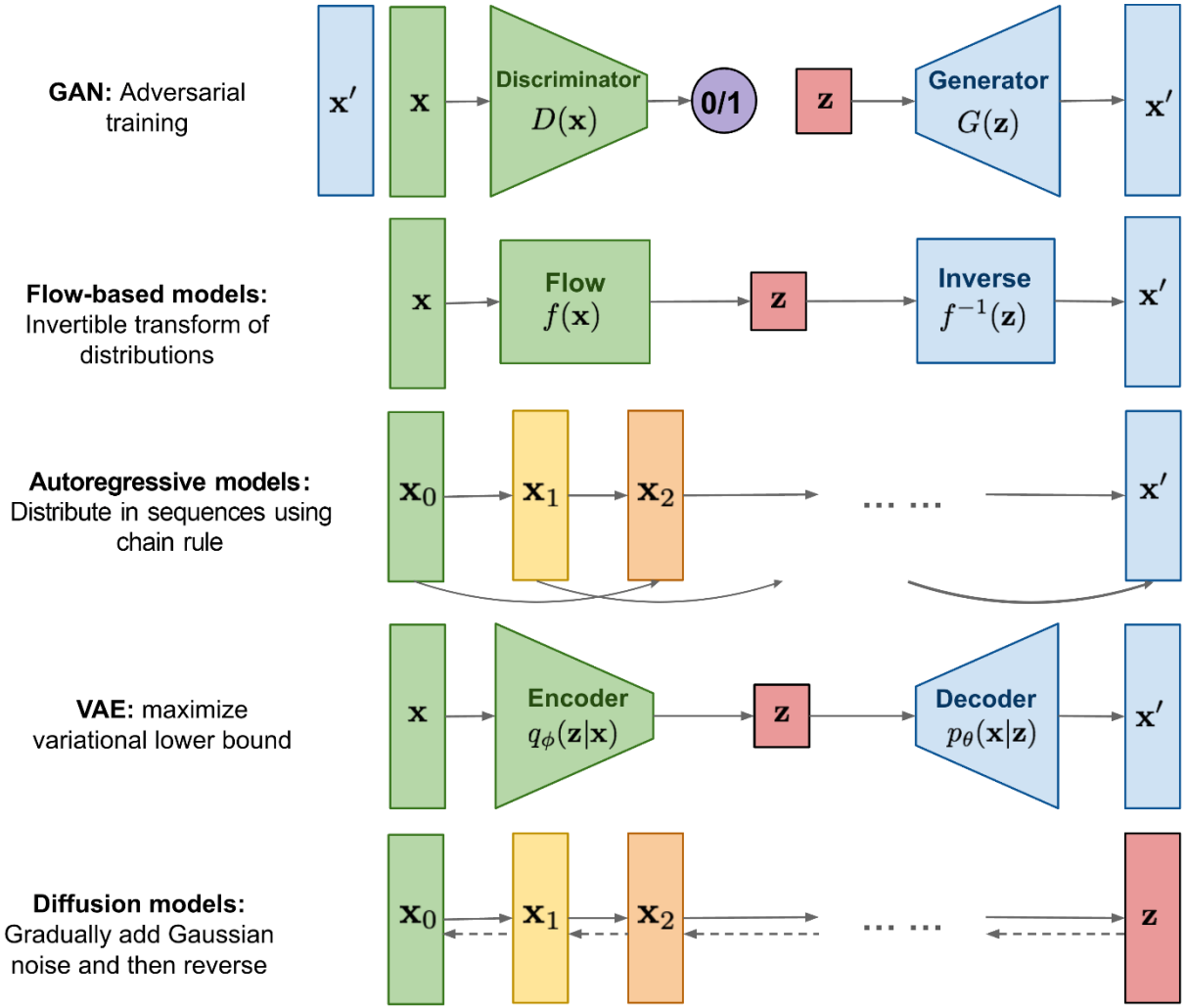


Backup

References

- A. Ciprijanovic et al., “DeepAstroUDA: Semi-Supervised Universal Domain Adaptation for Cross-Survey Galaxy Morphology Classification and Anomaly Detection”, Mach. Learn. Sci. Tech. 4 (2023) 025013, arXiv:2302.02005.
- CMS Collaboration, “Search for resonant production of strongly-coupled dark matter in proton-proton collisions at 13 TeV”, JHEP 06 (2022) 156, arXiv:2112.11125.
- CMS Collaboration, “Search for dark QCD with emerging jets in proton-proton collisions at √s = 13 TeV”, JHEP 07 (2024) 142, arXiv:2403.01556.
- F. Canelli, A. de Cosa, L. Le Pottier, J. Niedziela, K. Pedro, M. Pierini, “Autoencoders for Semivisible Jet Detection”, JHEP 02 (2022) 074, arXiv:2112.02864.
- CMS Collaboration, “Wasserstein normalized autoencoder”, CMS-PAS-MLG-24-002.
- K. Pedro and P. Shyamsundar, “Optimal Mass Variables for Semivisible Jets”, SciPost Phys. Core 6 (2023) 067, arXiv:2303.16253.
- CMS Collaboration, “Level-1 Trigger Calorimeter Image Convolutional Anomaly Detection Algorithm", CMS-DP-2023-086.
- CMS Collaboration, “Anomaly Detection in the CMS Global Trigger Test Crate for Run 3", CMS-DP-2023-079.
- S. Banerjee et al., “Denoising Convolutional Networks to Accelerate Detector Simulation”, J. Phys. Conf. Ser. 2438 (2023) 012079, arXiv:2202.05320.
- O. Amram and K. Pedro, “Denoising diffusion models with geometry adaptation for high fidelity calorimeter simulation”, Phys. Rev. D 108 (2023) 072014, arXiv:2308.03876.
- C. Jiang, S. Qian, and H. Qu, “Choose Your Diffusion: Efficient and flexible ways to accelerate the diffusion model in fast high energy physics simulation”, arXiv:2401.13162.
- C. Krause et al., “CaloChallenge 2022: A Community Challenge for Fast Calorimeter Simulation”, arXiv:2410.21611.
- J. Duarte et al., “FPGA-accelerated machine learning inference as a service for particle physics computing”, Comp. Soft. Big Sci. 3 (2019) 13, arXiv:1904.08986.
- D. Rankin et al., “FPGAs-as-a-Service Toolkit (FaaST)”, Proc. H2RC (2020) 38, arXiv:2010.08556.
- M. Wang, T. Yang, et al., “GPU-accelerated machine learning inference as a service for computing in neutrino experiments”, Front. Big Data 3 (2021) 604083, arXiv:2009.04509.
- J. Krupa, K. Lin, et al., “GPU coprocessors as a service for deep learning inference in high energy physics”, Mach. Learn. Sci. Tech. 2 (2021) 035005, arXiv:2007.10359.
- T. Cai et al., “Accelerating Machine Learning Inference with GPUs in ProtoDUNE Data Processing”, Comp. Soft. Big Sci. 7 (2023) 11, arXiv:2301.04633.
- C. Savard et al., “Optimizing High Throughput Inference on Graph Neural Networks at Shared Computing Facilities with the NVIDIA Triton Inference Server”, Comp. Soft. Big Sci. 8 (2024) 14, arXiv:2312.06838.
- A. Gunny et al., “Hardware-accelerated Inference for Real-Time Gravitational-Wave Astronomy,” Nature Astron. 6 (2022) 529, arXiv:2108.12430.
- Y.-T. Chou et al., “ACTS as a Service”, ACAT 2024.
- B. Riedel, “IceSONIC - Network AI Inference on Coprocessors for IceCube Offline Processing”, Fast ML Coprocessor Meeting.

Generative Models

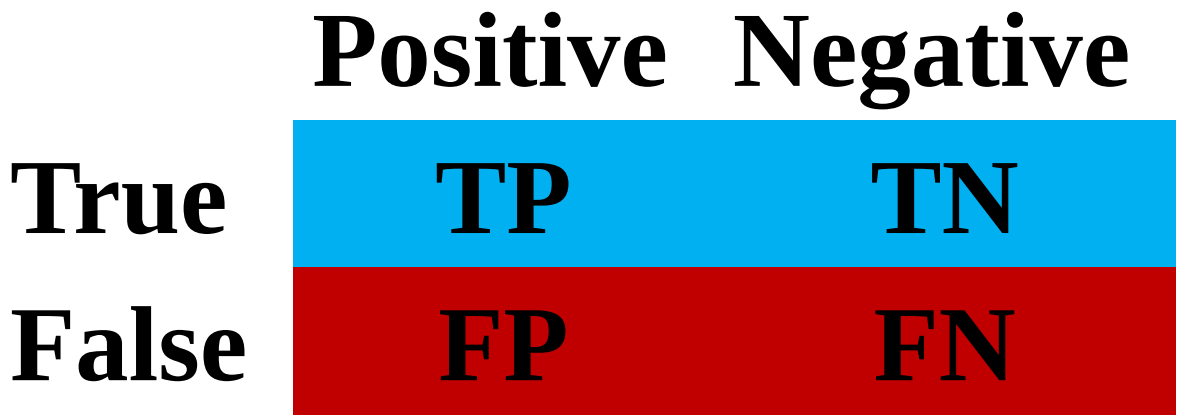


L. Weng

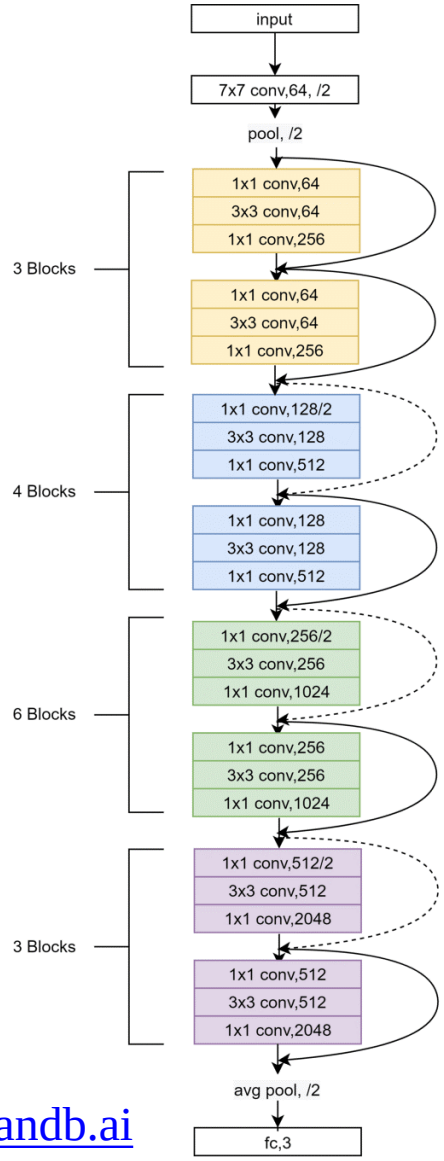
- *Implicit* density estimation: Generative Adversarial Networks (GANs)
  - Pros: fast
  - Cons: can suffer from mode collapse, lack of convergence, etc.
- *Exact* density estimation: Normalizing Flows (NFs), Autoregressive models (ARs)
  - Pros: accurate, fast in one direction
  - Cons: poor scaling, slow in other direction
- *Approximate* density estimation: Variational Autoencoders (VAEs), Diffusion Models (DMs)
  - VAEs: fast, but limited quality
  - DMs: high quality, but slow
- *Non-generative*: reweighting, refinement
  - Classification- or regression-based

Classifier Metrics

- Receiver Operator Characteristic (ROC): TP vs. FP
  - AUC: Area Under (ROC) Curve
- Accuracy = (TP + TN) / (TP + TN + FP + FN)
- Precision = TP / (TP + FP)
- Recall = TP / (TP + FN)
- F1 score = 2·(precision·recall)/(precision + recall)

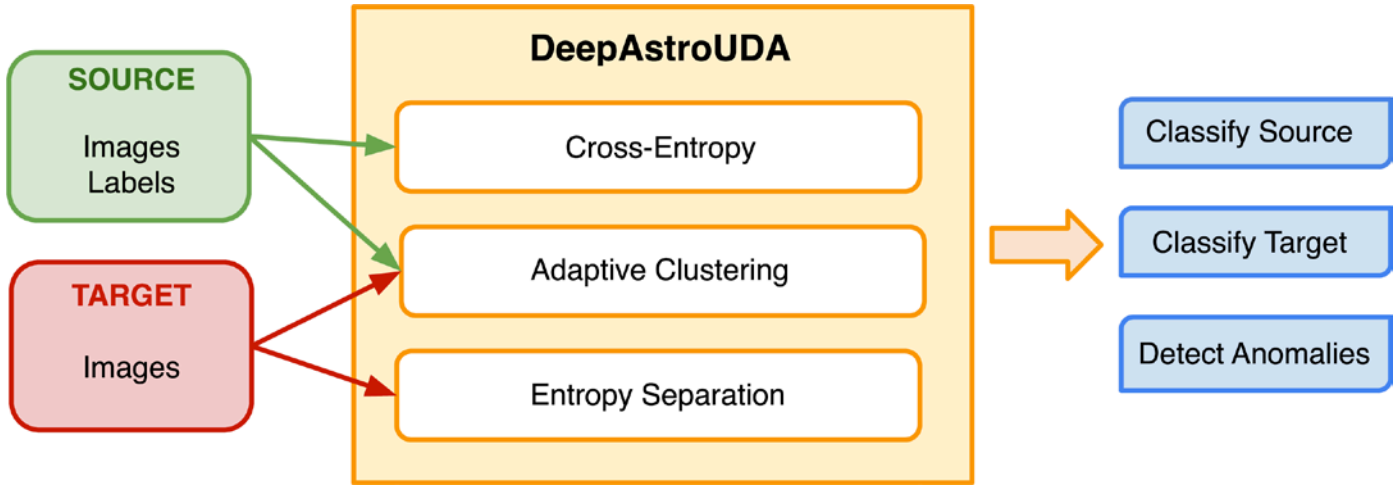


DeepAstroUDA Architecture



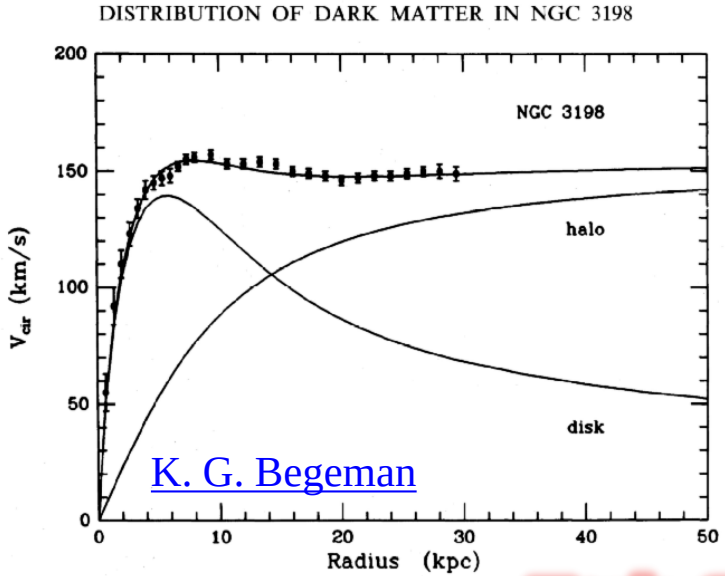
wandb.ai

- Backbone: ResNet50 with skip connections
- Training setup:
  - Labeled and unlabeled inputs passed to different loss terms



Evidence for Dark Matter

Anomalous Galactic Rotation Curves

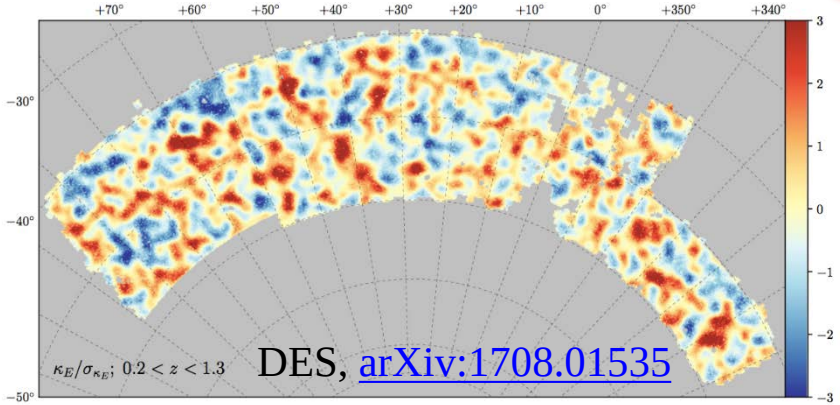


K. G. Begeman

Bullet Cluster Collision

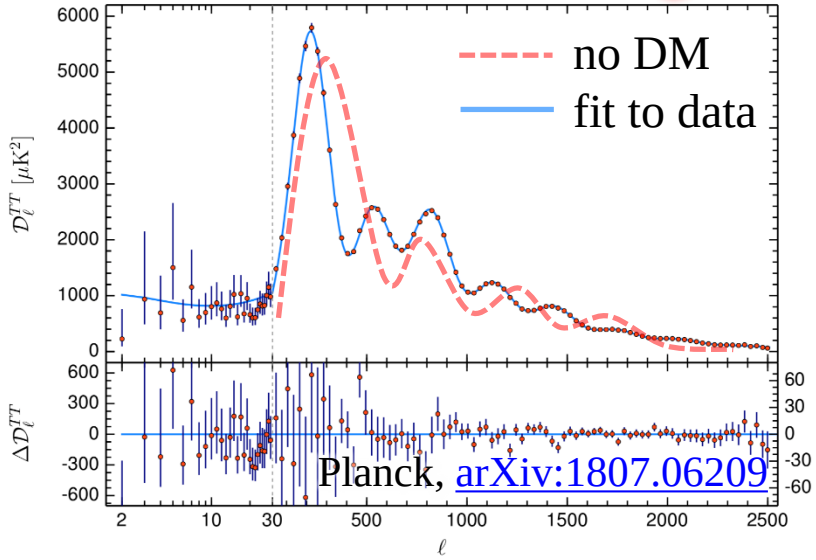
NASA, arXiv:astro-ph/0608407

Weak Gravitational Lensing



DES, arXiv:1708.01535

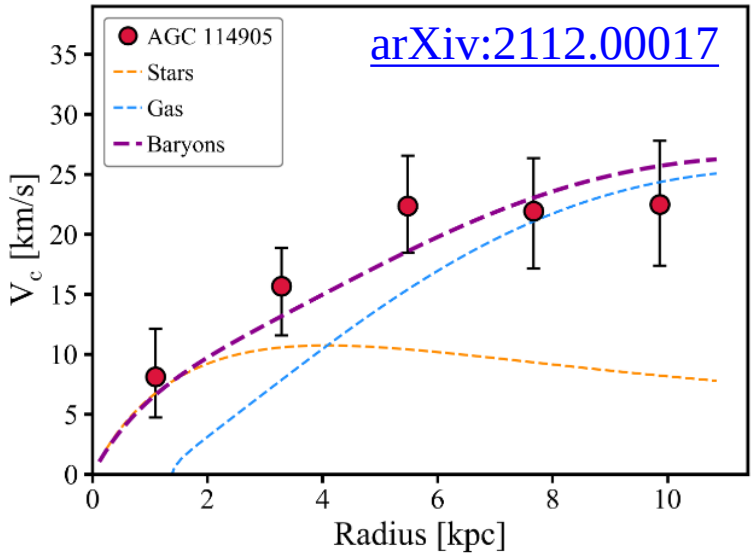
Cosmic Microwave Background



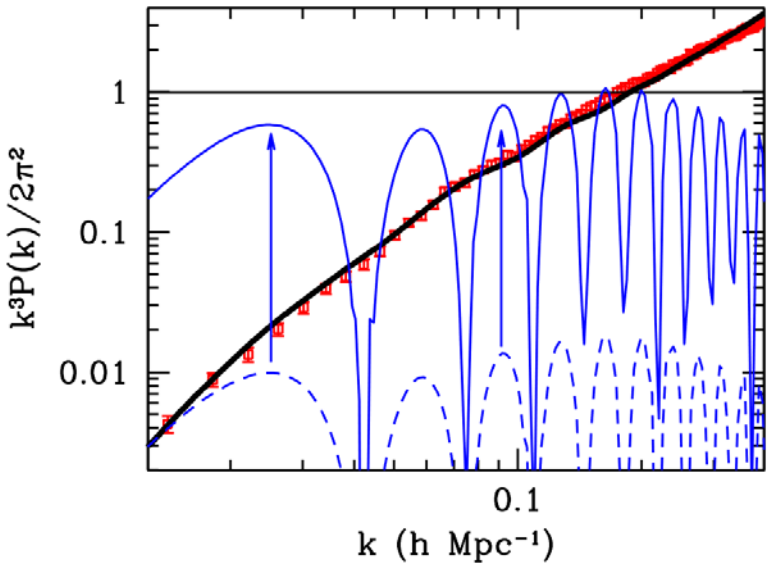
Planck, arXiv:1807.06209

More Evidence for Dark Matter

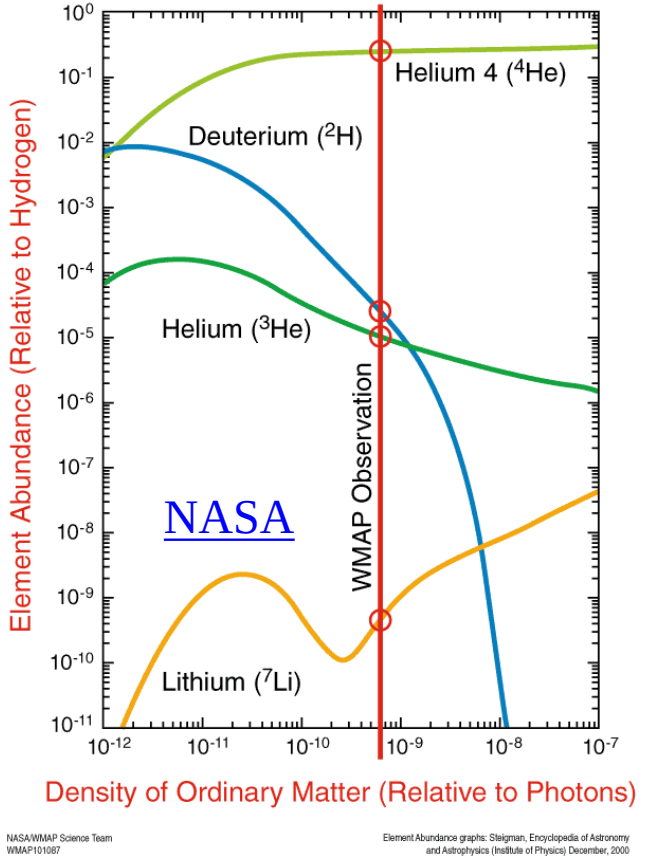
Non-Anomalous Galactic Rotation Curves



Matter Power Spectrum

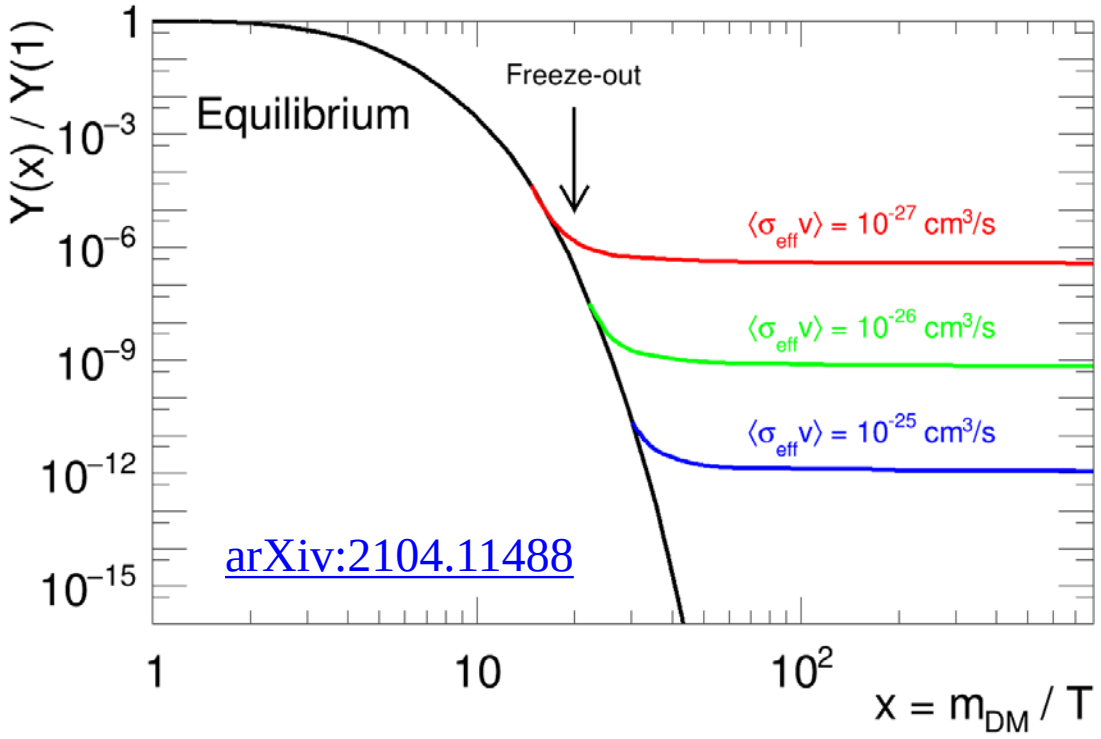


Light Element Abundance in Big Bang Nucleosynthesis

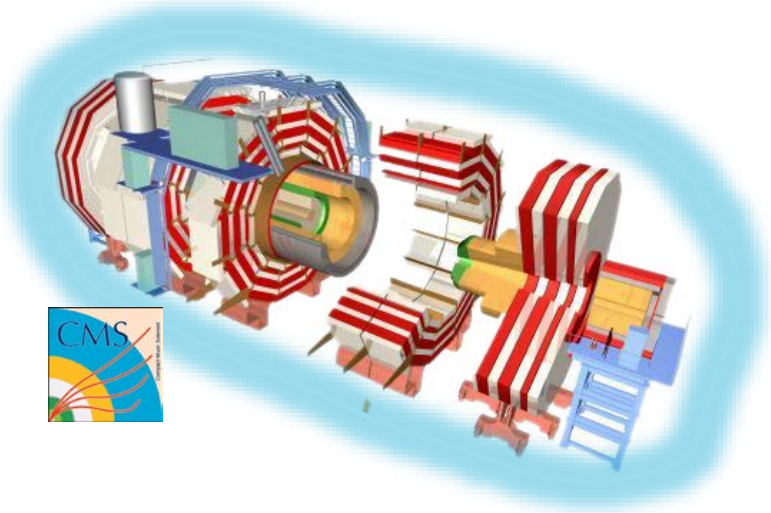


Relic Density

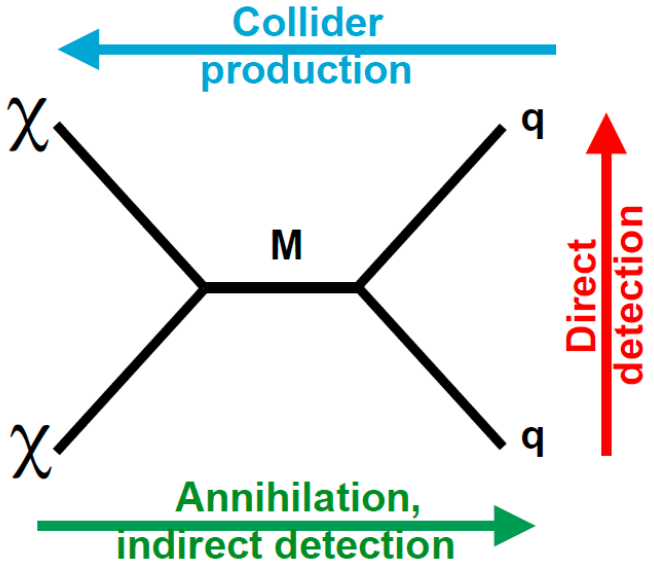
- Assuming:
  - o "Cold" dark matter (massive particles, moving slowly)
  - o At thermal equilibrium with the universe
- Number density of dark matter evolves vs. time (/temperature)
  - o "Freeze out" occurs when Hubble expansion exceeds DM annihilation rate $\langle\sigma_{eff}v\rangle$
    - ➢ Constant abundance called "relic density"
  - o Smaller coupling → earlier freeze-out → higher relic abundance
- From WMAP and Planck: $\Omega_c h^2 \approx 0.12$

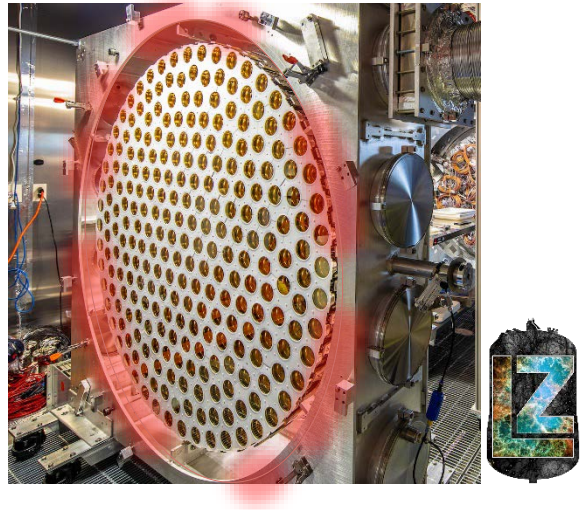


Dark Matter Searches

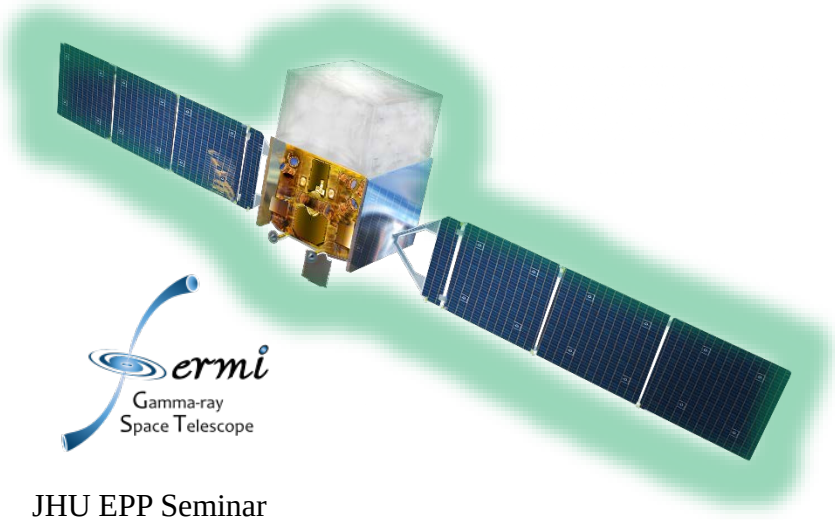


stronger for lighter DM and spin-dependent interactions, as well as many extended dark sector models





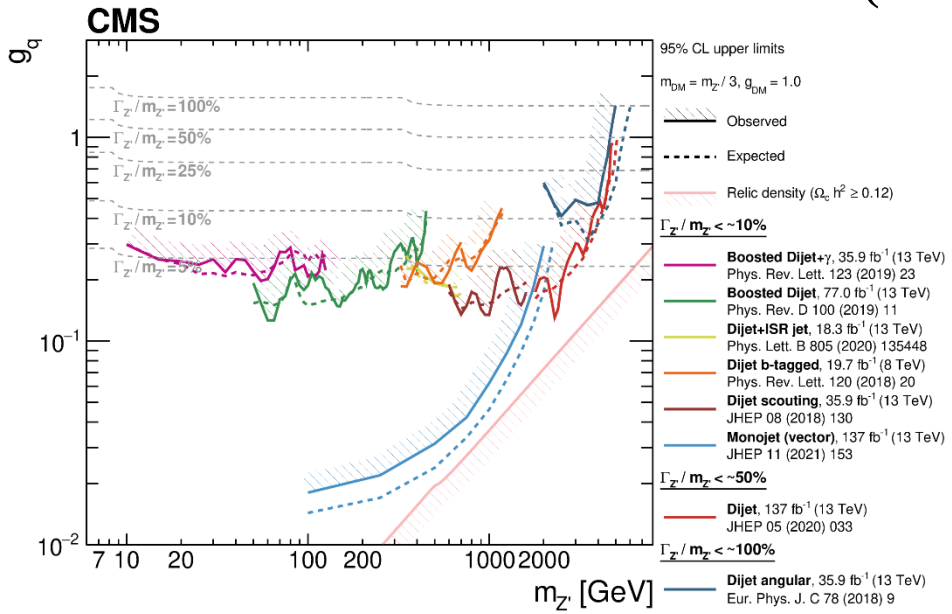
sensitive to small scattering cross sections, but struggles with DM-lepton interactions, very light DM, composite DM

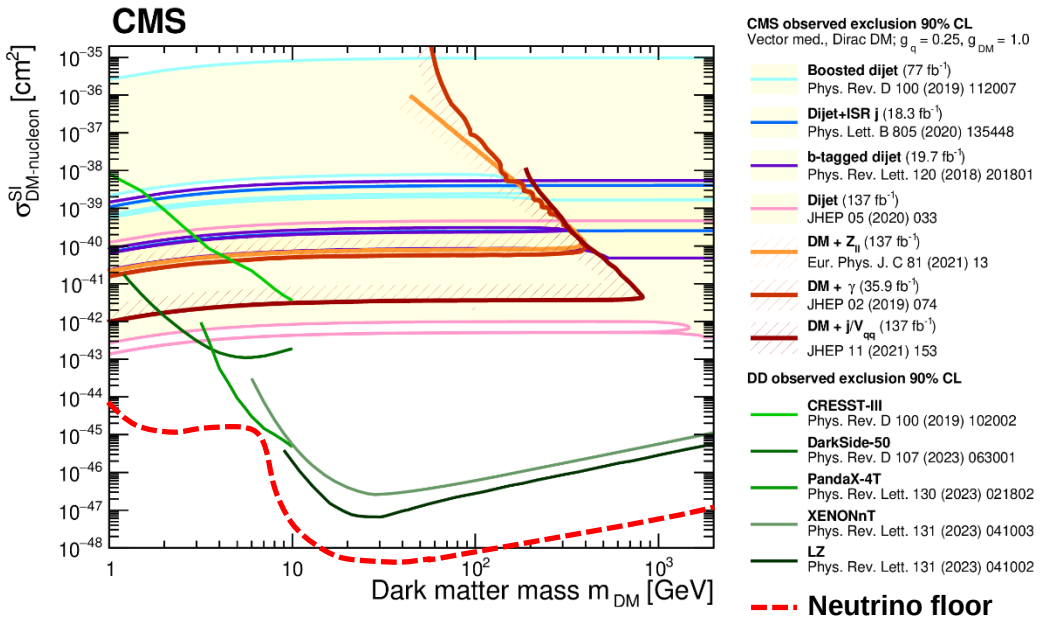


probes relic density and DM-SM couplings, but need to model astrophysical backgrounds and DM density profile

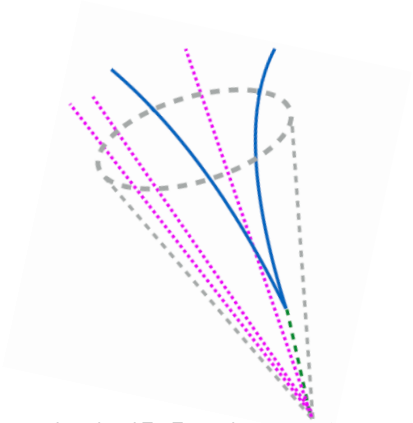
Collider DM Searches

- Collider DM searches target events with large p_T^{miss}, focused on "simplified models":
  - Fermionic dark matter particle χ with mass m_{DM} (invisible) → WIMP (weakly interacting massive particle)
  - Mediator with mass m_{med}, couplings g_q and g_χ
- LHC Run 2 exclusions:
  - g_q excluded over multiple decades of Z' mass, almost to range allowed by relic density
  - Direct detection searches provide best spin-independent cross section exclusion, except at low mass
    - Almost reaches "neutrino floor" (σ^{SI} similar to σ_ν)

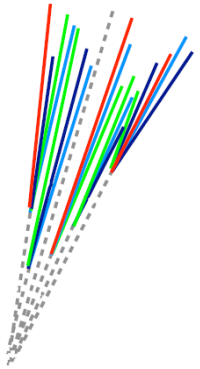

CMS
95% CL upper limits
$m_{DM} = m_{Z'}/3$, $g_{DM} = 1.0$
Observed
Expected
Relic density ($\Omega_c h^2 \geq 0.12$)
$\Gamma_{Z'}/m_{Z'} = 100\%$
$\Gamma_{Z'}/m_{Z'} = 50\%$
$\Gamma_{Z'}/m_{Z'} = 25\%$
$\Gamma_{Z'}/m_{Z'} = 10\%$
$\Gamma_{Z'}/m_{Z'} = 5\%$
$\Gamma_{Z'}/m_{Z'} < \sim 10\%$
Boosted Dijet+γ, 35.9 fb^{-1} (13 TeV) Phys. Rev. Lett. 123 (2019) 23
Boosted Dijet, 77.0 fb^{-1} (13 TeV) Phys. Rev. D 100 (2019) 11
Dijet+ISR jet, 18.3 fb^{-1} (13 TeV) Phys. Lett. B 805 (2020) 135448
Dijet b-tagged, 19.7 fb^{-1} (8 TeV) Phys. Rev. Lett. 120 (2018) 20
Dijet scouting, 35.9 fb^{-1} (13 TeV) JHEP 08 (2018) 130
Monojet (vector), 137 fb^{-1} (13 TeV) JHEP 11 (2021) 153
$\Gamma_{Z'}/m_{Z'} < \sim 50\%$
Dijet, 137 fb^{-1} (13 TeV) JHEP 05 (2020) 033
$\Gamma_{Z'}/m_{Z'} < \sim 100\%$
Dijet angular, 35.9 fb^{-1} (13 TeV) Eur. Phys. J. C 78 (2018) 9
g_q
$m_{Z'}$ [GeV]


CMS
$\sigma^{SI}_{DM\text{-}nucleon}$ [cm^2]
Dark matter mass m_{DM} [GeV]
CMS observed exclusion 90% CL
Vector med., Dirac DM; $g_q = 0.25$, $g_{DM} = 1.0$
Boosted dijet (77 fb^{-1}) Phys. Rev. D 100 (2019) 112007
Dijet+ISR j (18.3 fb^{-1}) Phys. Lett. B 805 (2020) 135448
b-tagged dijet (19.7 fb^{-1}) Phys. Rev. Lett. 120 (2018) 201801
Dijet (137 fb^{-1}) JHEP 05 (2020) 033
DM + Z_{ll} (137 fb^{-1}) Eur. Phys. J. C 81 (2021) 13
DM + γ (35.9 fb^{-1}) JHEP 02 (2019) 074
DM + j/V_{qq} (137 fb^{-1}) JHEP 11 (2021) 153
DD observed exclusion 90% CL
CRESST-III Phys. Rev. D 100 (2019) 102002
DarkSide-50 Phys. Rev. D 107 (2023) 063001
PandaX-4T Phys. Rev. Lett. 130 (2023) 021802
XENONnT Phys. Rev. Lett. 131 (2023) 041003
LZ Phys. Rev. Lett. 131 (2023) 041002
Neutrino floor

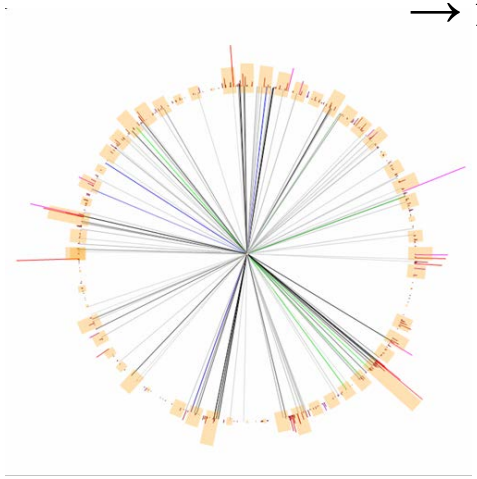
Phenomenology



Semivisible jets (SVJs)
mixture of stable and unstable dark hadrons
→ p_T^{miss} aligned with jets



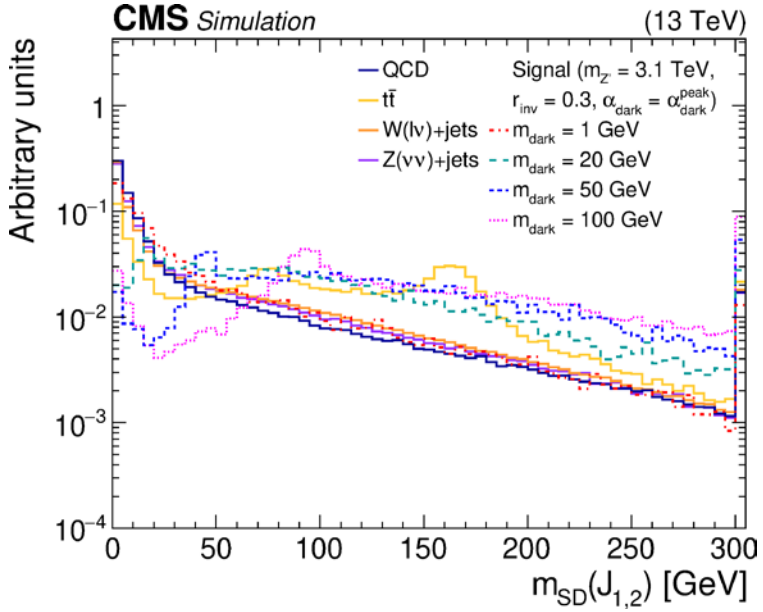
Emerging jets (EMJs)
dark hadrons decay after some lifetime
→ multiple displaced vertices within each jet

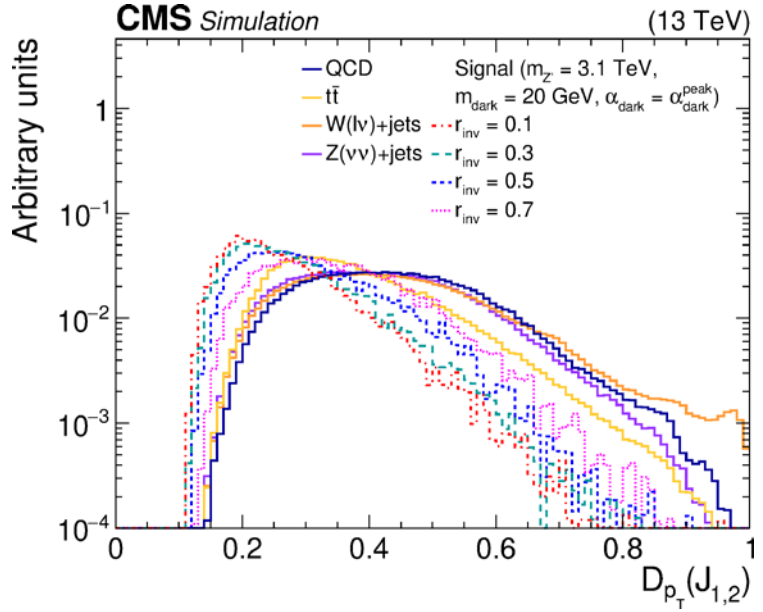


Soft unclustered energy patterns (SUEPs)
expand to confining theories with large ‘t Hooft coupling,
beyond QCD-like regime → spherical distribution of low-p_T tracks

Tagging Semivisible Jets

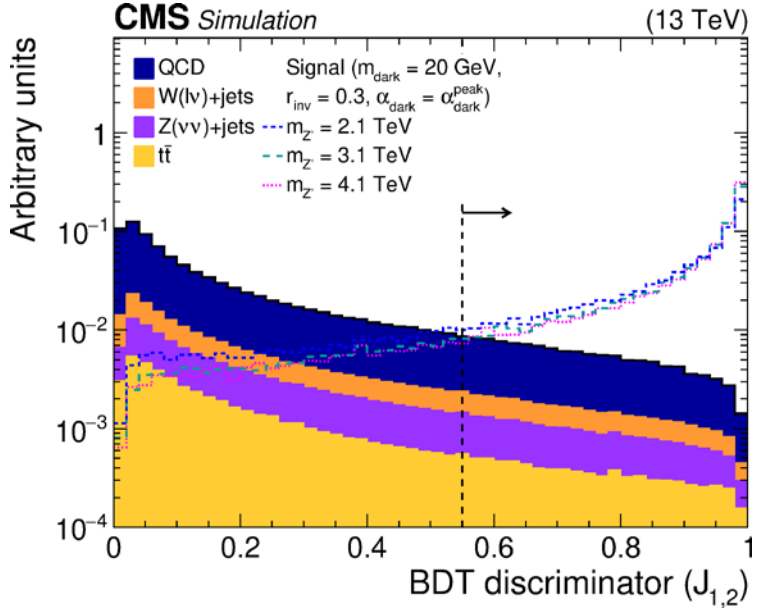
- Various jet substructure variables (& $\Delta\varphi(J, p_T^{miss})$) can weakly discriminate between semivisible jets and SM background jets
  - Heavy object tagging: m_{SD}, τ_{21}, τ_{32}, $N_2^{(1)}$, $N_3^{(1)}$
  - Quark-gluon discrimination: D_{p_T}, σ_{major}, σ_{minor}, girth
  - Flavor (energy fractions): f_γ, $f_{h^\pm}$, f_{h0}, f_e, f_μ
- ➢ Combine useful variables into a BDT for strong discrimination!
  - Background: equal mix of QCD and $t\bar{t}$; signal: mix of many models
  - Reweight background jet p_T spectrum to match signal: avoid sculpting



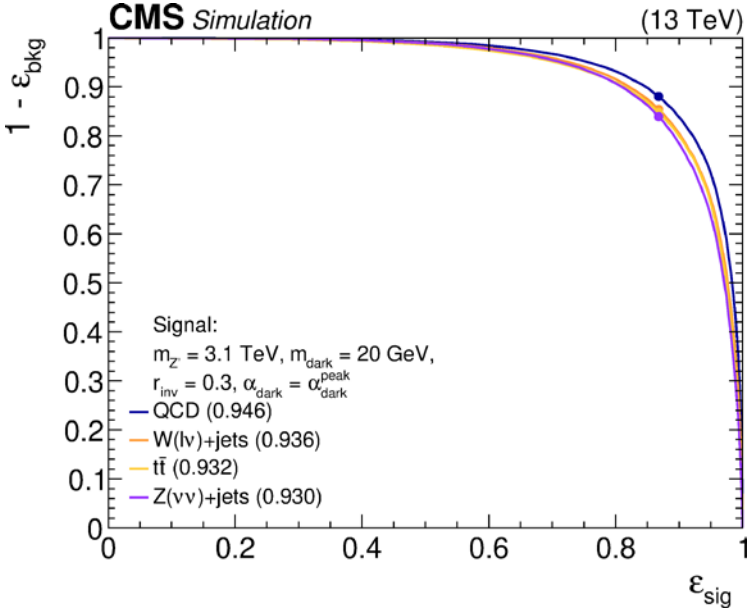


SVJ Tagger Performance

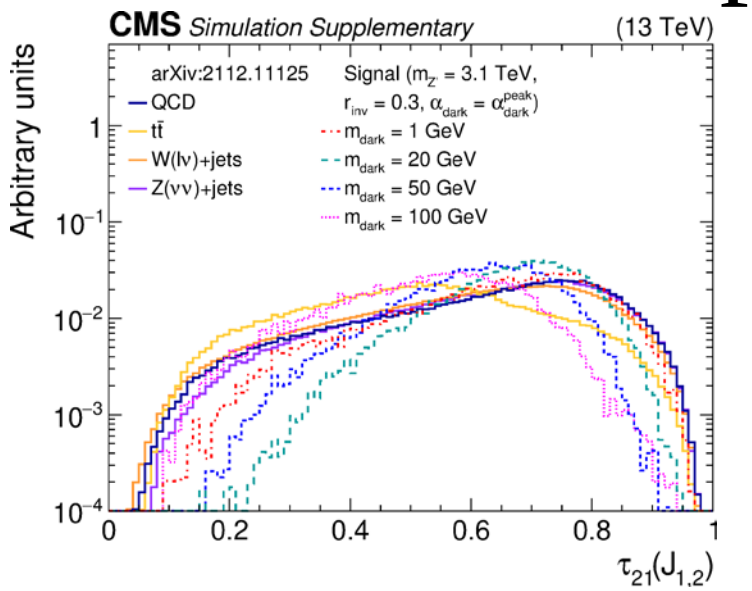
	$m_{Z'}$ = 3.1 TeV, m_{dark} = 20 GeV, r_{inv} = 0.3, $\alpha_{dark} = \alpha_{dark}^{peak}$		
	Acc (WP = 0.5)	AUC	$1/\varepsilon_B$ (ε_S = 0.3)
QCD	0.881	0.947	651.4
$t\bar{t}$	0.881	0.931	270.6
W(ℓν)+jets	0.881	0.936	441.5
Z(νν)+jets	0.881	0.930	420.7

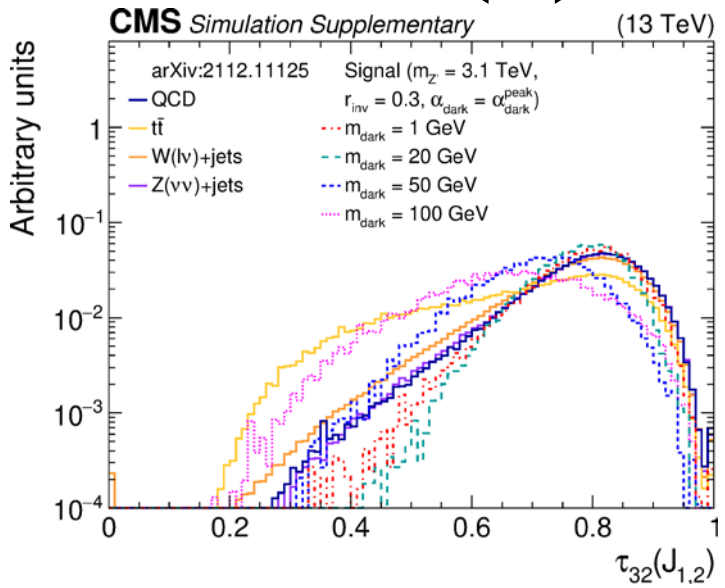


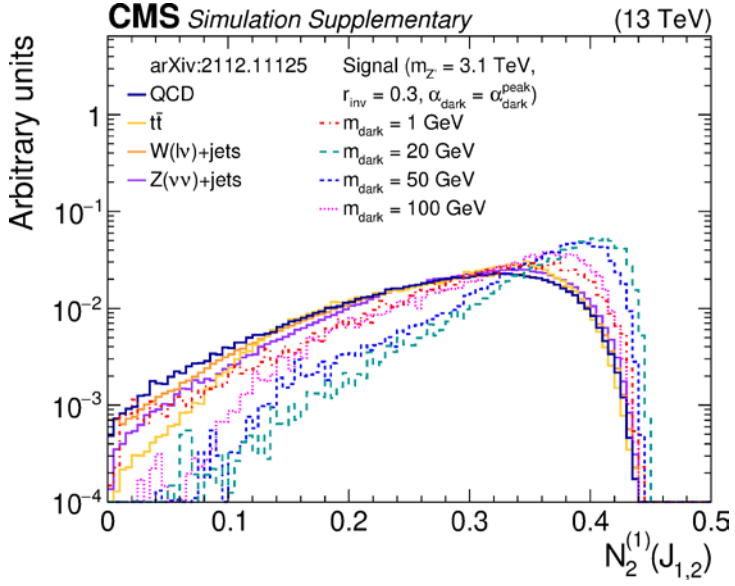
- Strong and consistent performance
  - Training on only QCD ($t\bar{t}$) caused misclassification of $t\bar{t}$ (QCD) jets at rate of 10–20%
  - Some inefficiency for signals with high or low m_{dark}
- Working point 0.55 chosen based on background estimation

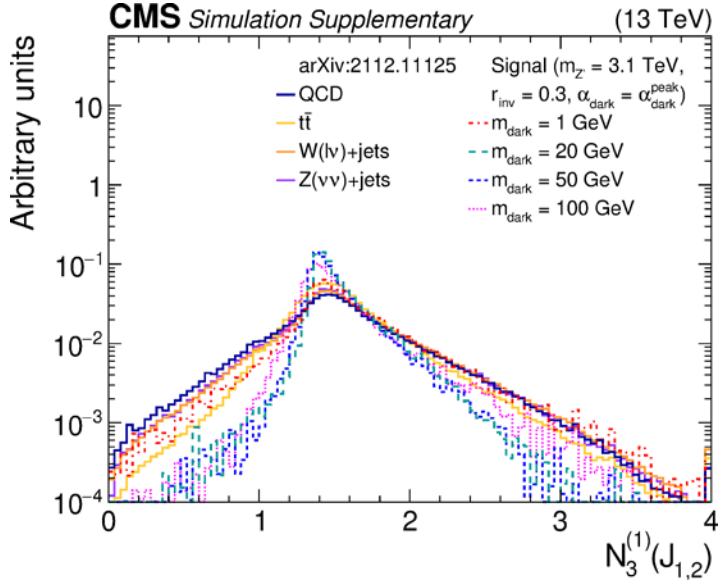


SVJ BDT Input Variables (1)

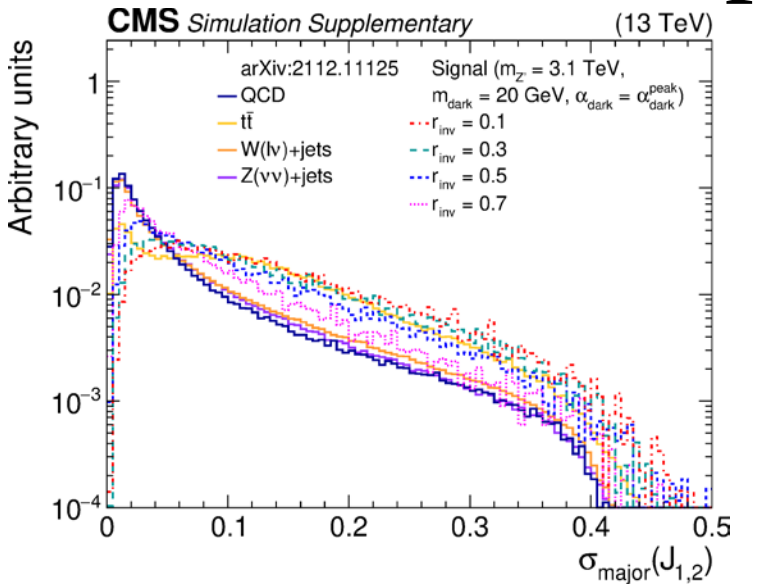


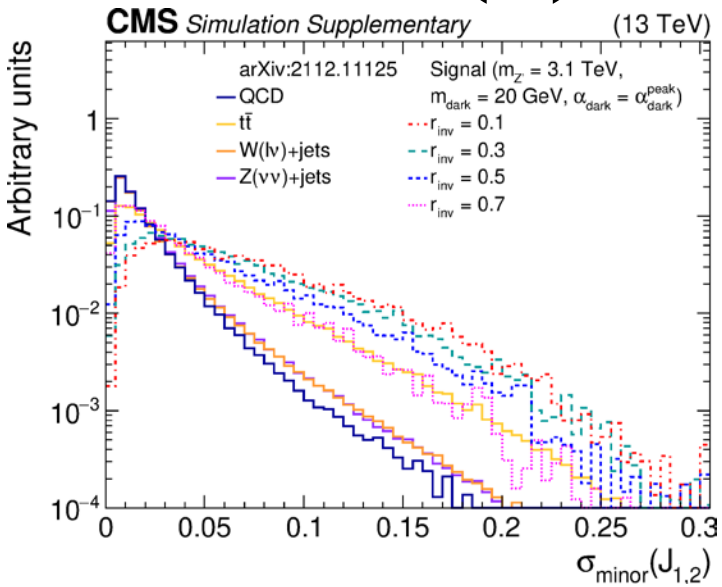


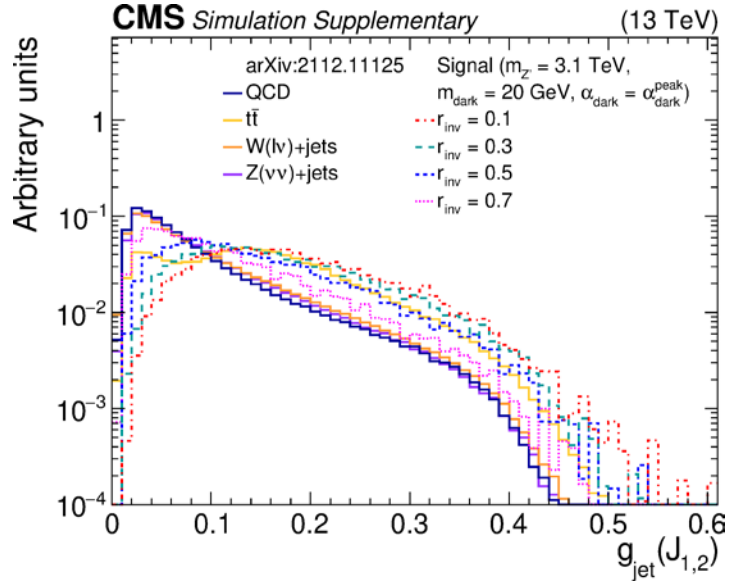


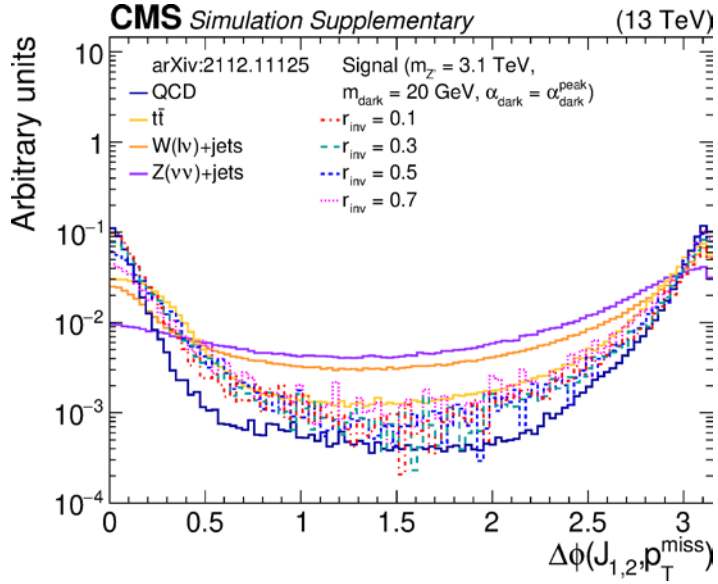


SVJ BDT Input Variables (2)

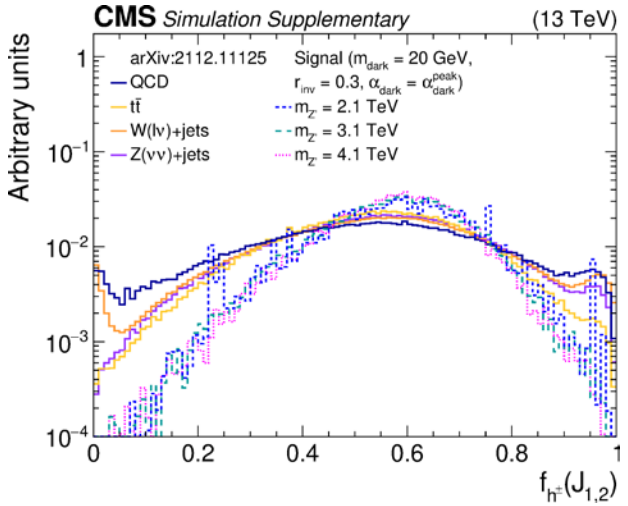


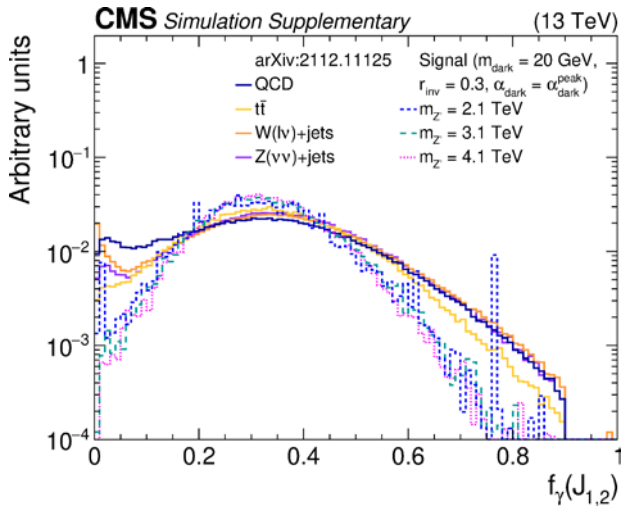


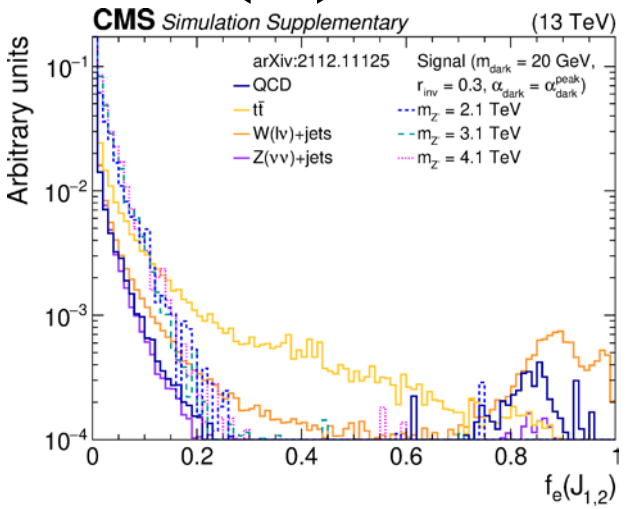


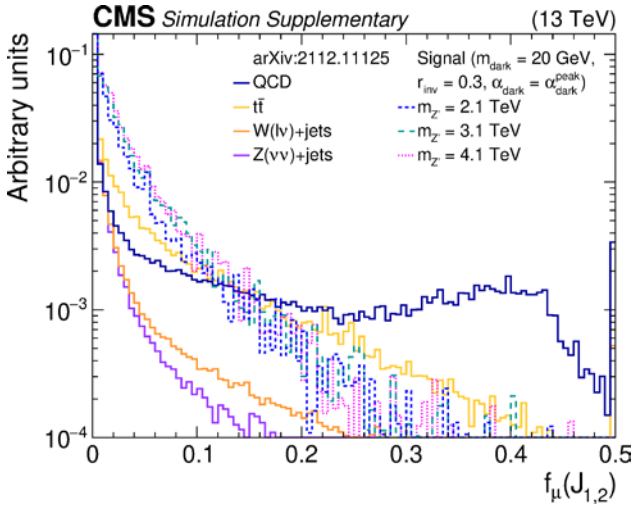


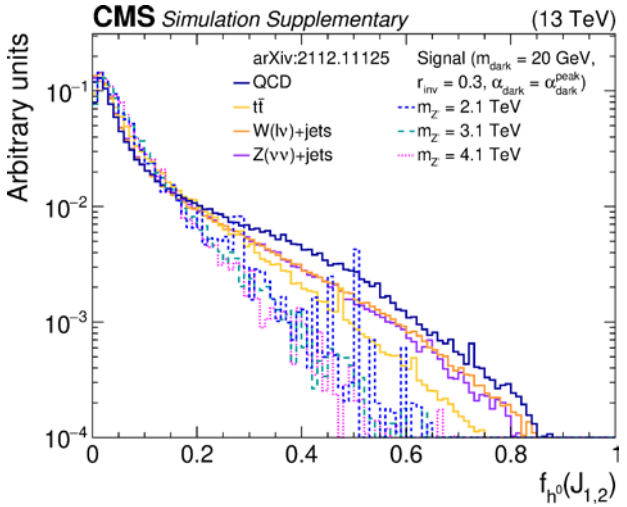
SVJ BDT Input Variables (3)



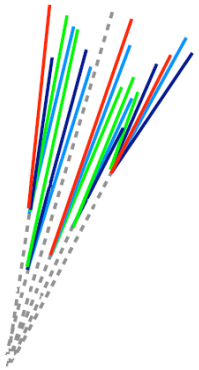




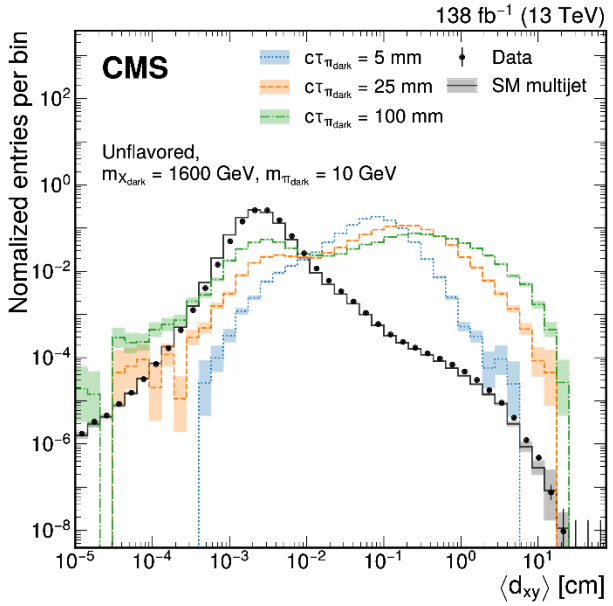


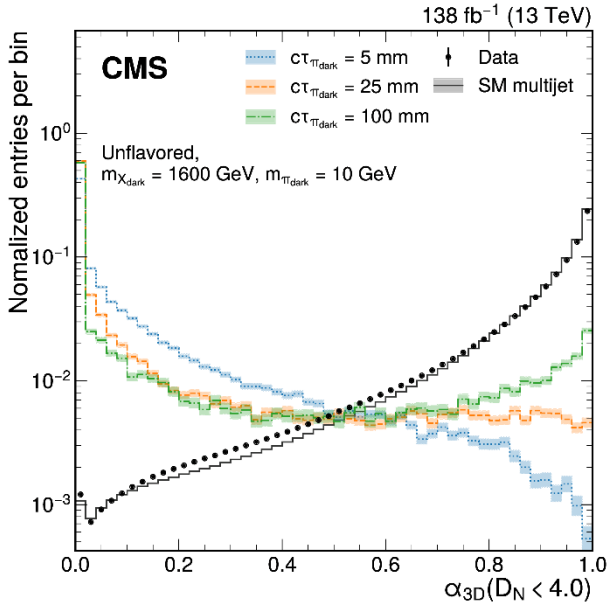


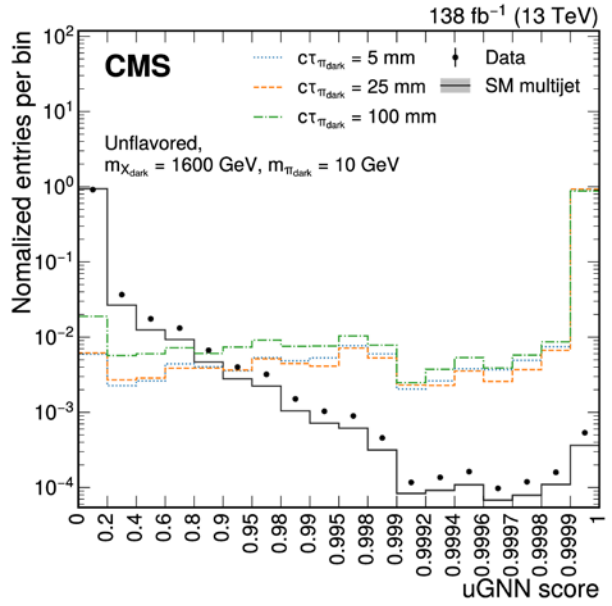
How to Identify EMJs



- Vertex reconstruction inefficient – few tracks with few tracker hits
- Focus on track displacement and related quantities, such as:
  - ‹d_{xy}›: median transverse impact parameter for tracks associated to the jet
  - α_{3D}: fraction of p_T from prompt tracks (w/ impact parameter significance $D_N < 4.0$)
- Alternative approach:
  - Train ParticleNet (dynamic graph convolution network) using momentum and track displacement from all tracks associated with a jet

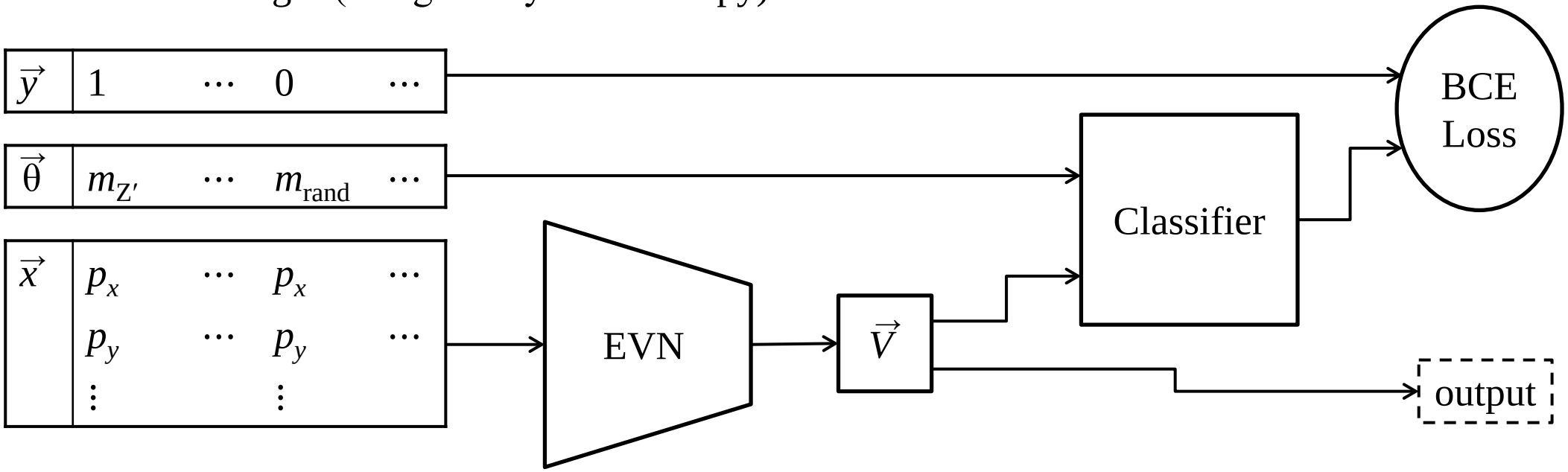






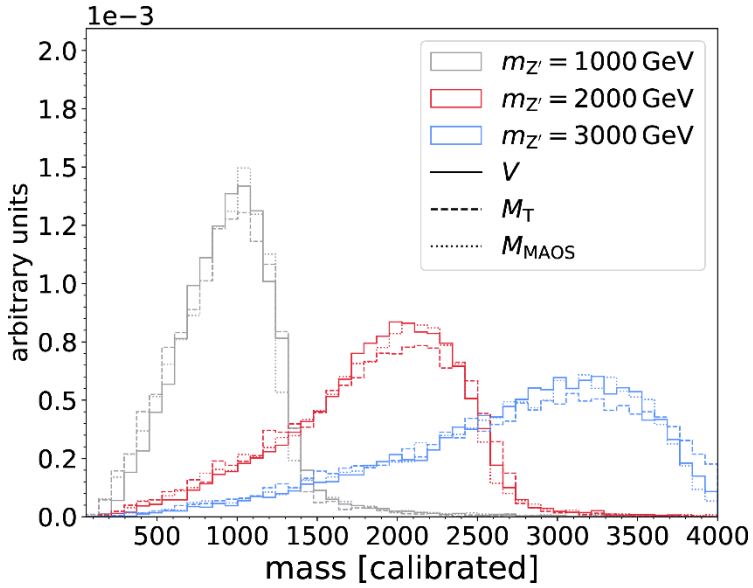
Interpretable, Semi-Supervised ML

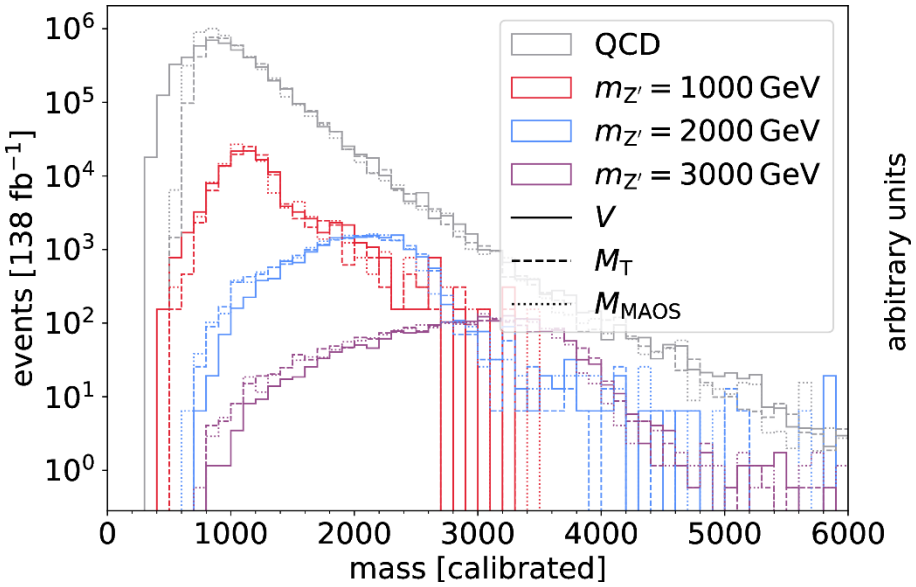
- Rejecting background is important… but can we learn more about signal?
- Derive new $\vec{V}(\vec{x})$ from inputs $\vec{x}$ to maximize *mutual information* with model parameter(s) $\vec{\theta}$
  - *Not* a regression: learns an actual, generalized function of inputs, e.g. approx. invariant mass *function*
- Both components are simple fully-connected networks (few layers)
  - Classifier uses $\vec{V}$ (from Event Variable Network bottleneck) to distinguish events w/ correct $\vec{\theta}$ from events w/ wrong $\vec{\theta}$ (using binary crossentropy)

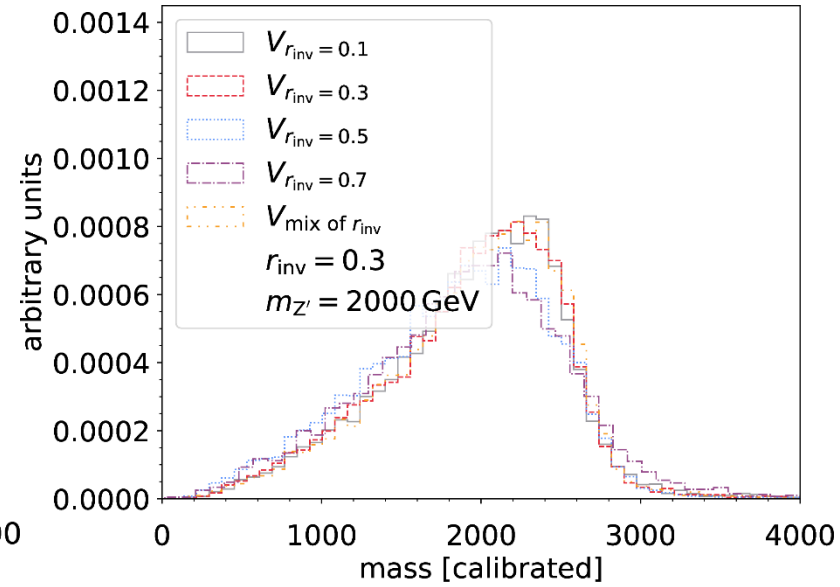


EVN for SVJ

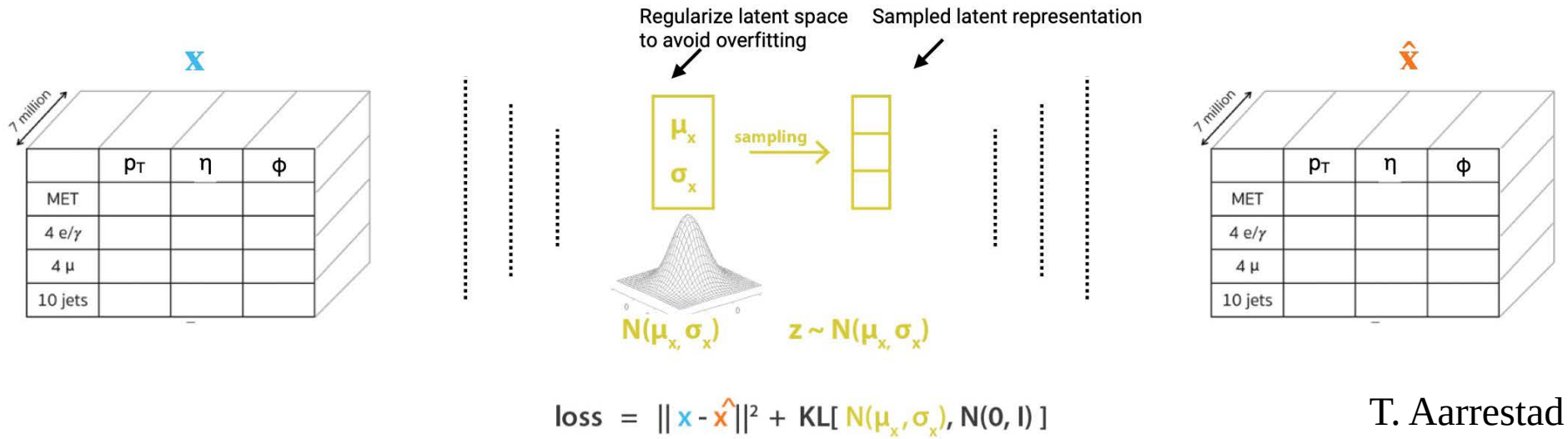
- Can improve on M_T for Z′ mass reconstruction (similar to M_{T2}-assisted on shell algorithm)
  - Definitive answer to question posed back in 2015!
- Falling distribution for background, though trained only on signal
  - Facilitates bump hunt approach
- Generalizes well, even across r_{inv}







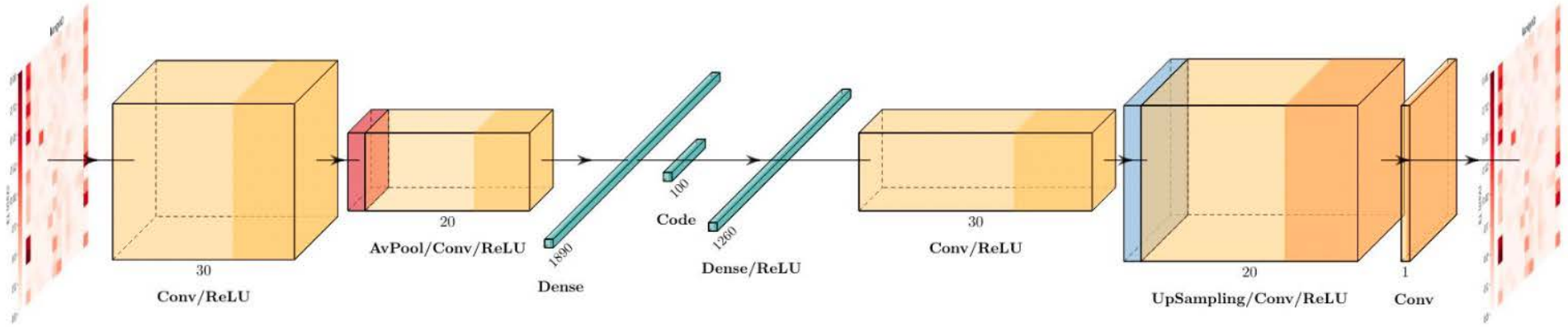
AXOL1TL

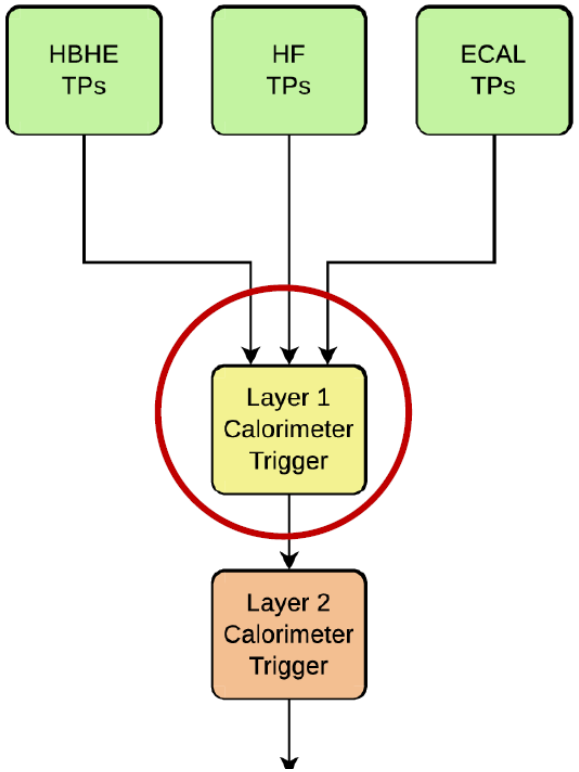


T. Aarrestad

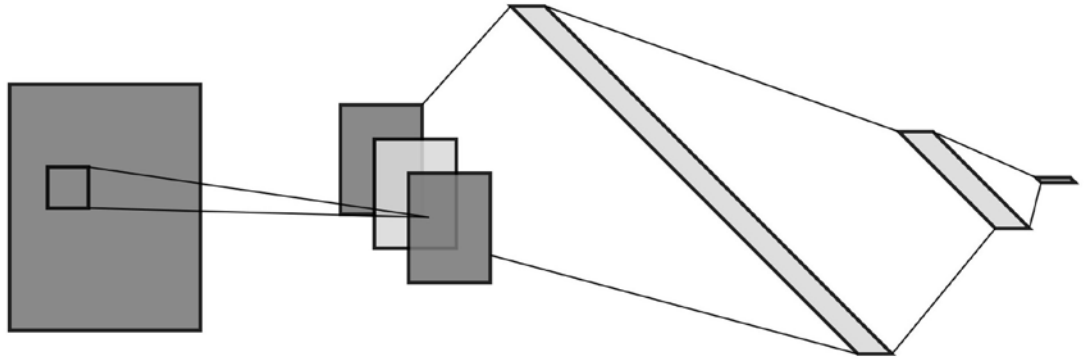
- K-L divergence expands to ~ $\mu^2 + \sigma^2 - 1 - \log(\sigma^2)$
- Dominated by just μ^2 → use this as score instead of full reconstruction error
- Drop second half of network (decoder) for inference → substantially reduces latency on FPGA (50 ns)

CICADA





- Input is 2D map of energy deposited in calorimeters (ECAL + HCAL)
- Initial model (above, “teacher”): 300K parameters
- Deployed model (below, “student”): 3K parameters
  - Trained to reproduce teacher model using knowledge distillation
  - ~100 ns latency on FPGA

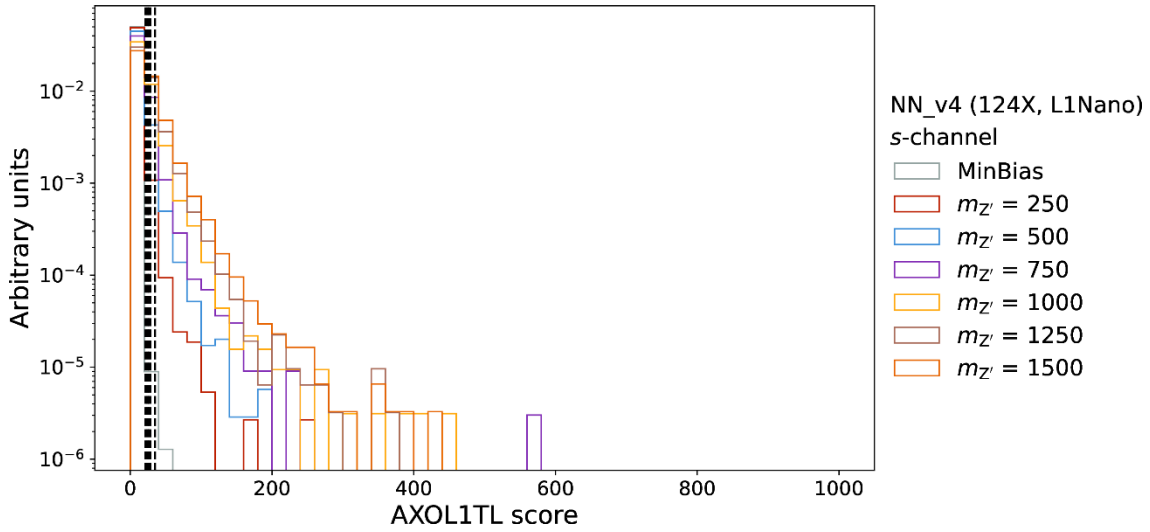


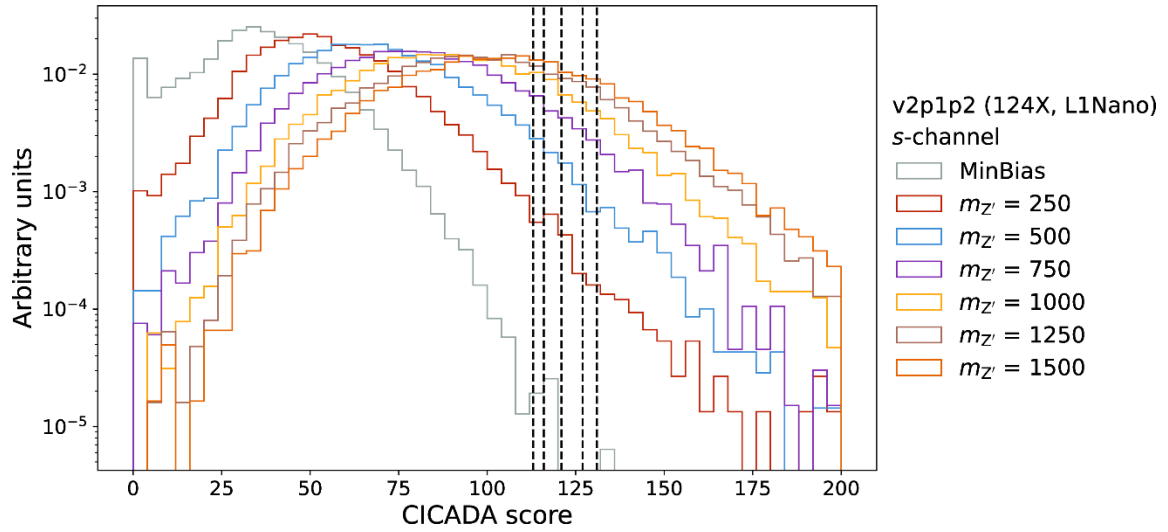
Anomaly Trigger Distributions

AXOL1TL

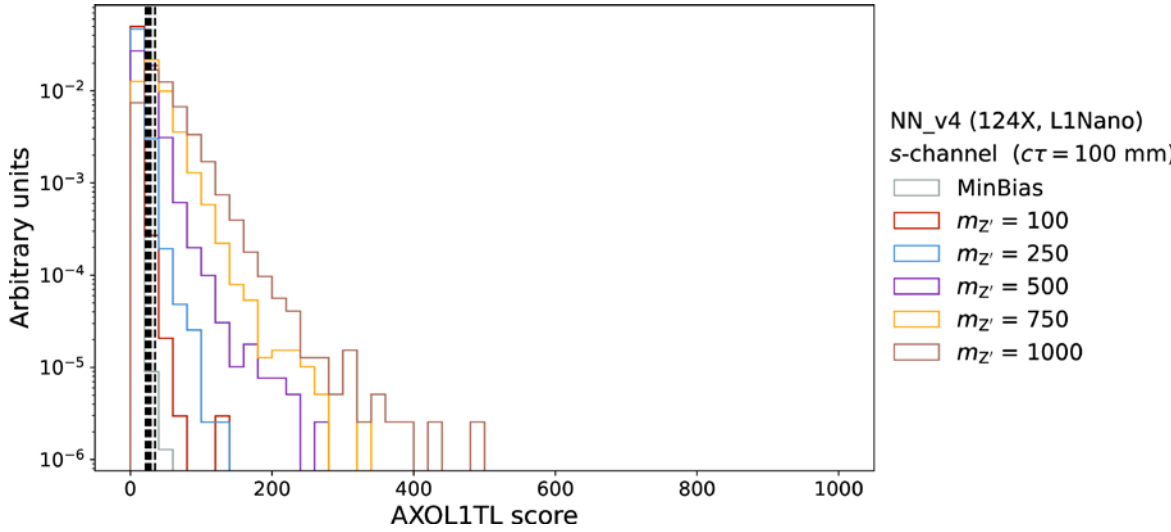
CICADA

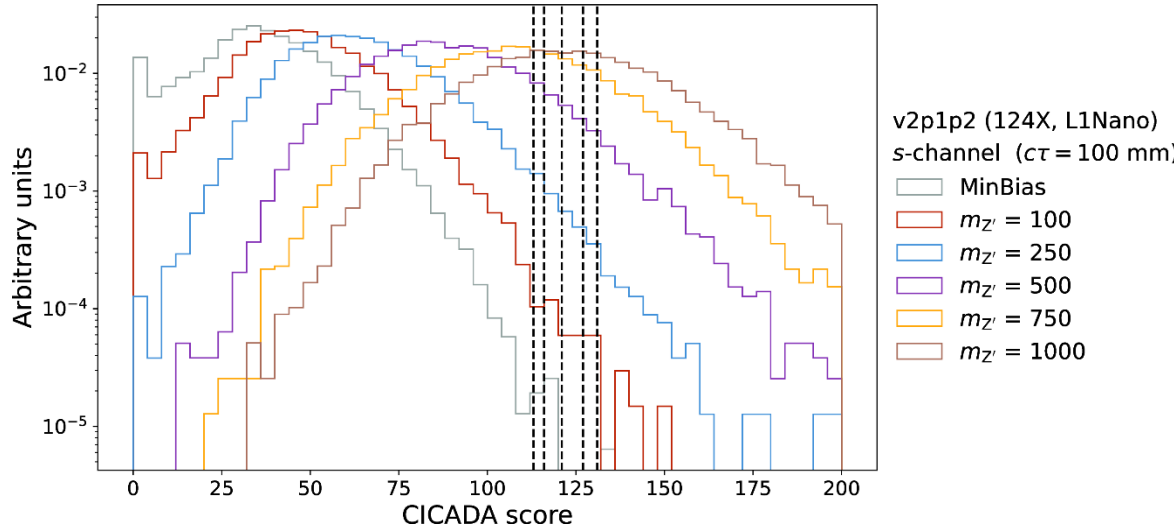
SVJ





EMJ

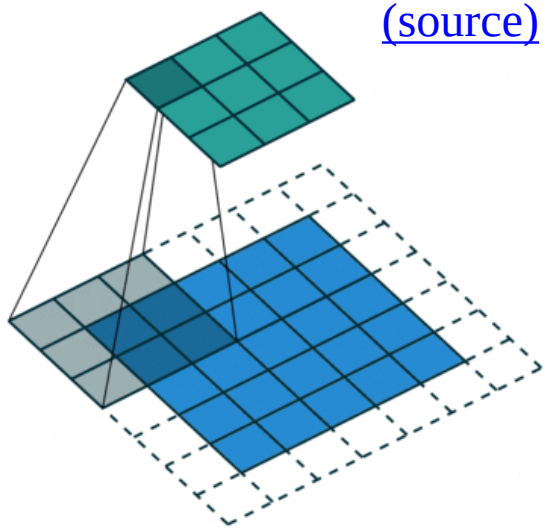




Why Convolutions?

- Convolutions started the modern machine learning revolution (AlexNet, 2012)
  - *Spatial locality* and translational invariance
  - Shared weights → fewer parameters, *better scaling*
  - Highly *efficient* on GPUs: spatial locality implies memory locality
- Ideally suited for computer vision with rectangular images
  - Application to irregular geometries requires innovations
- Graph neural networks?
  - **Pro**: natural representation for irregular geometries
  - **Cons**: adjacency matrices consume substantial memory; operations less local/efficient; hard to generate arbitrary output (masking technique exists, but difficult to scale)
- Point clouds or transformers?
  - **Pro**: no adjacency matrix consuming memory
  - **Con**: discards useful geometric information, which then must be learned from (often sparse) inputs

(source)



➢ For generative applications, convolutions still have a lot to offer!
  - And they can keep up with transformers when trained properly… arXiv:2310.16764

Generative Metrics

- Speed only matters if needed accuracy is achieved
  - Wrong answers can be obtained infinitely fast
- Looking at 1D histograms: not good enough!
  - Can miss high-dimensional correlations
- Best category: **integral probability metrics**

$$D_{\mathcal{F}}(p_{\text{real}}, p_{\text{gen}}) = \sup_{f \in \mathcal{F}} |\mathbb{E}_{\mathbf{x} \sim p_{\text{real}}} f(\mathbf{x}) - \mathbb{E}_{\mathbf{y} \sim p_{\text{gen}}} f(\mathbf{y})|$$

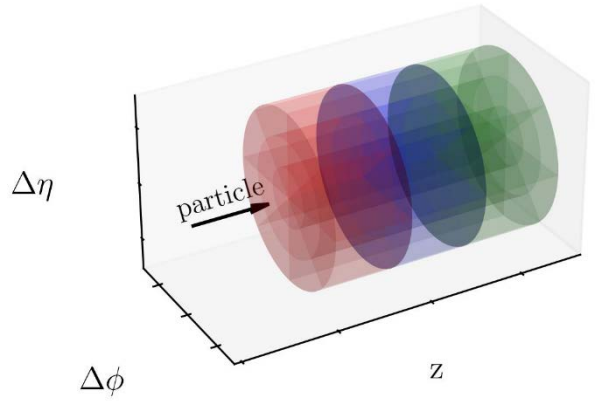
  - *Wasserstein distance* W_1: $\mathcal{F}$ is set of all K-Lipschitz functions
    - Only works well in 1D, biased in high-D
  - *Maximum mean discrepancy* (MMD): $\mathcal{F}$ is unit ball in reproducing kernel Hilbert space
    - Depends on choice of kernel
  - *Fréchet distance*: W_2 distance between Gaussian fits to (high-D) feature space
    - Features can be hand-engineered or obtained from NN activations
- Another interesting category: *classifier scores*
  - Train NN to distinguish real vs. generated
  - AUC score ranges from 0.5 to 1.0
- *Fréchet Particle Distance* most clearly distinguishes between two similar approaches (message passing GAN and generative adversarial particle transformer)

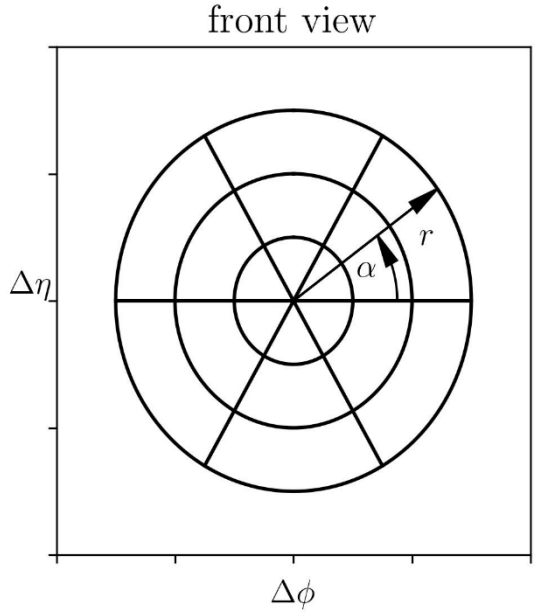
	FPD $\times 10^3$	KPD $\times 10^3$	W_1^M $\times 10^3$
Truth	0.08 ± 0.03	-0.006 ± 0.005	0.28 ± 0.05
MPGAN	$\mathbf{0.30 \pm 0.06}$	$\mathbf{-0.001 \pm 0.004}$	$\mathbf{0.54 \pm 0.06}$
GAPT	0.66 ± 0.09	0.001 ± 0.005	0.56 ± 0.08

arXiv:2211.10295

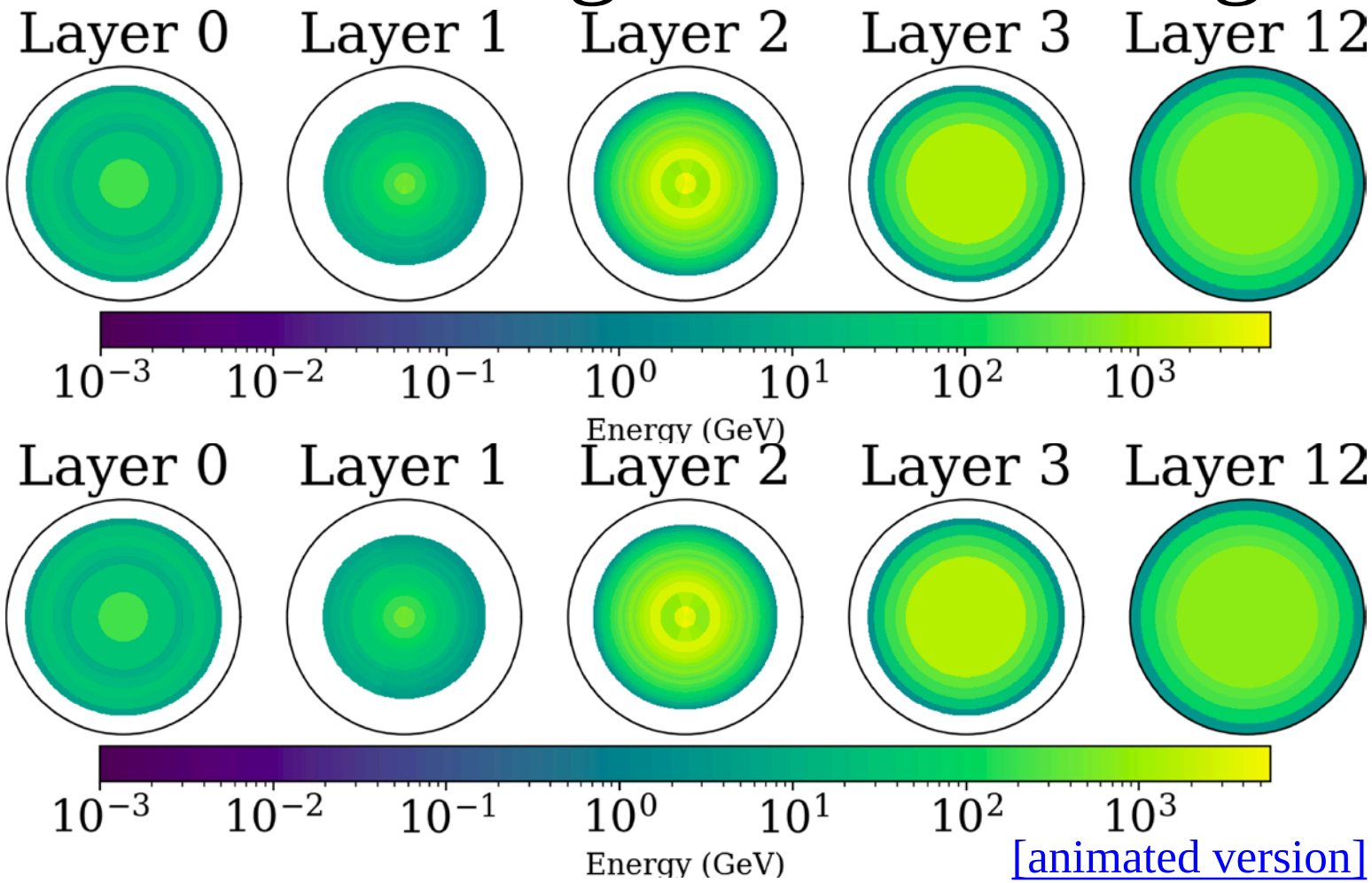
CaloChallenge Datasets

- CaloChallenge: common datasets for evaluation & comparison of generative models
  - Dataset 1: ATLAS calorimeter, irregular
    - Photons (368 voxels), 242K events
    - Pions (533 voxels), 241.6K events
  - Dataset 2: silicon-tungsten, 45 layers
    - Electrons (6480 voxels), 200K events
  - Dataset 3: silicon-tungsten, 45 layers
    - Electrons (40500 voxels), 200K events
- Preprocessing: (E_i = voxel energy)
  - Logit transform: $u_i = \log(x/_{1-x})$, $x \equiv \delta + (1 - 2\delta)E_i$
  - Standardization: $u_i' = (u_i - \bar{u})/\sigma_u$





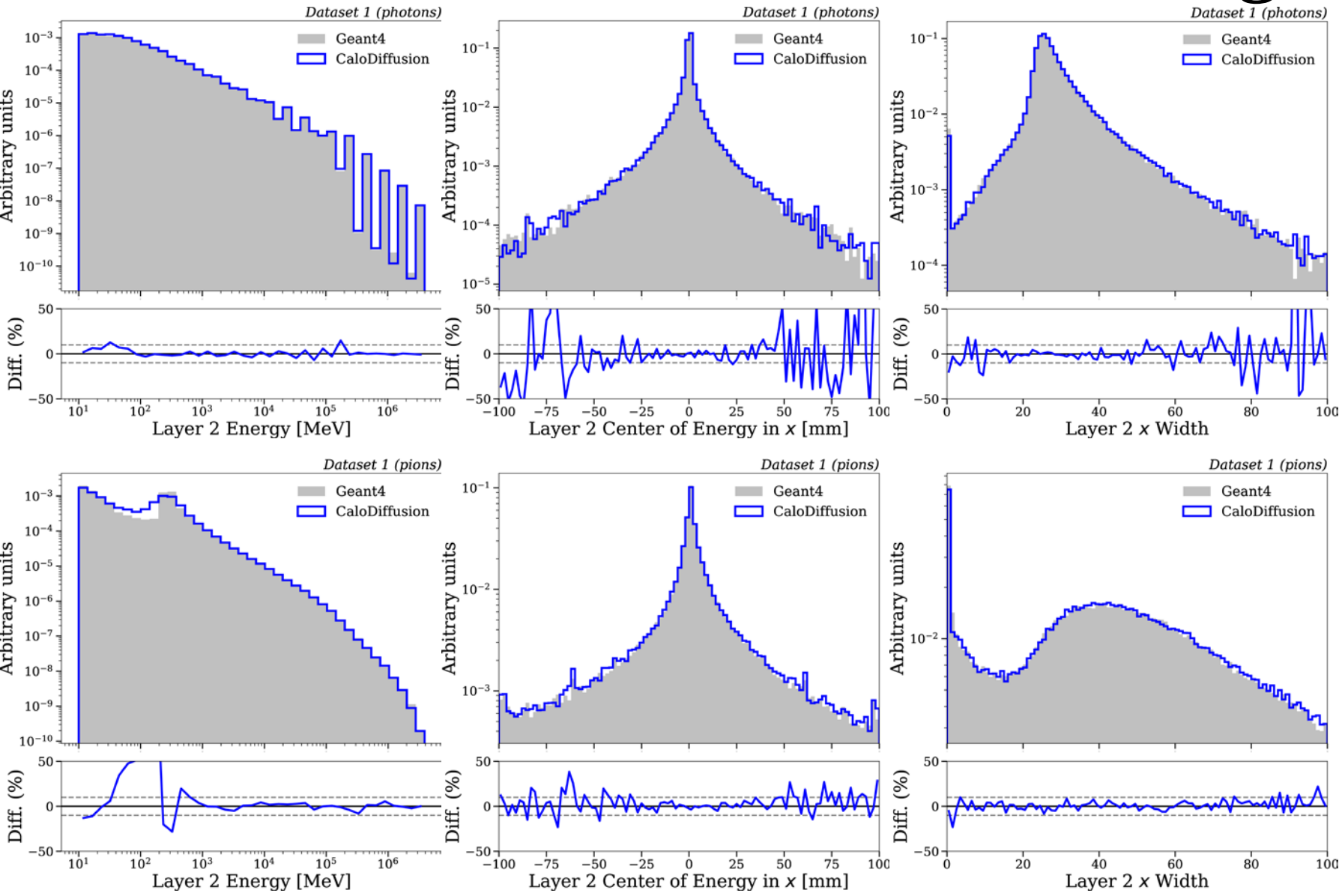
CaloDiffusion: Average CaloChallenge Showers



[animated version]

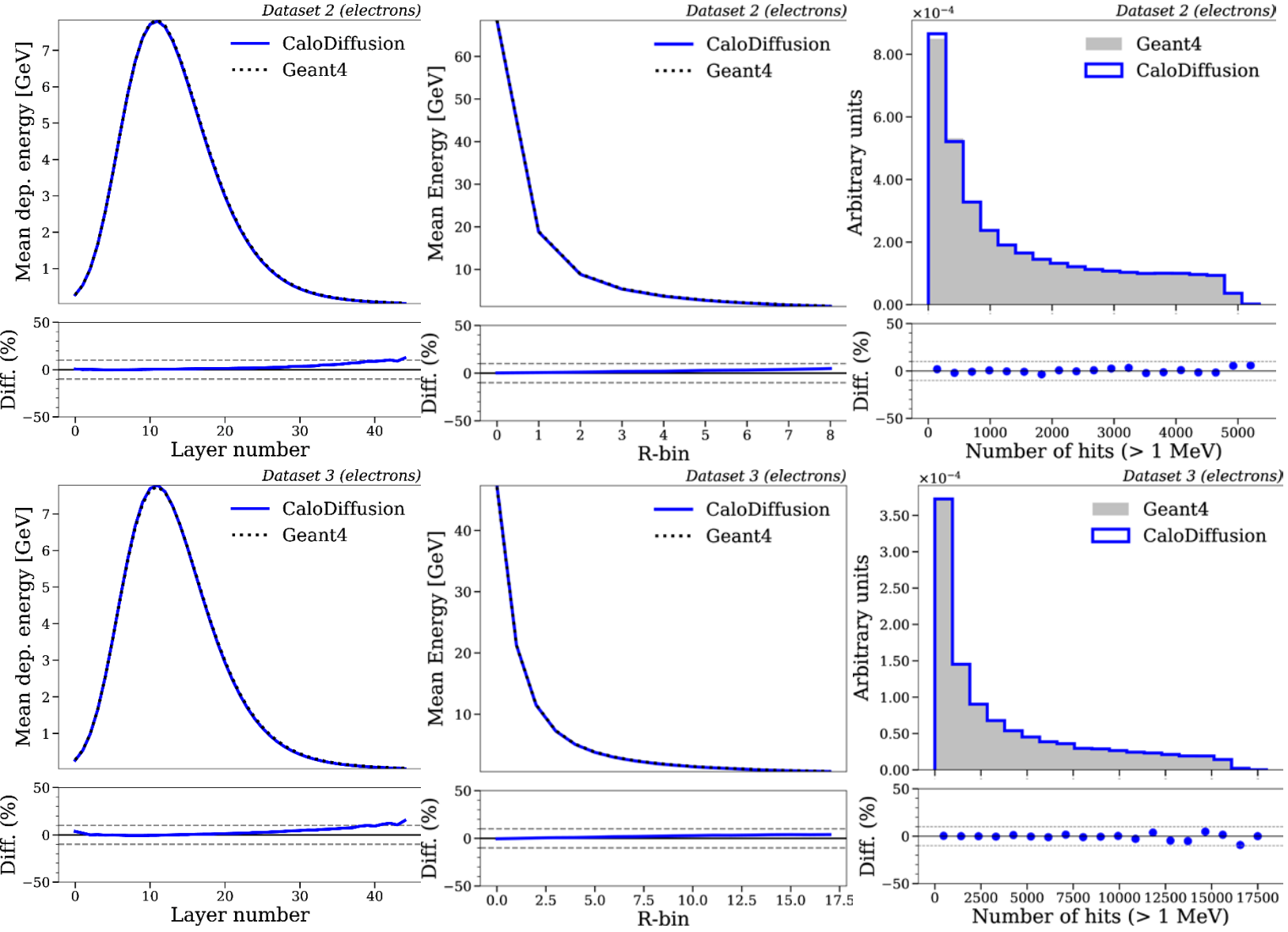
- Top: Geant4; bottom: CaloDiffusion (dataset 1, photons)
  - … or is it the other way around? Can you tell?

CaloDiffusion: CaloChallenge Dataset 1



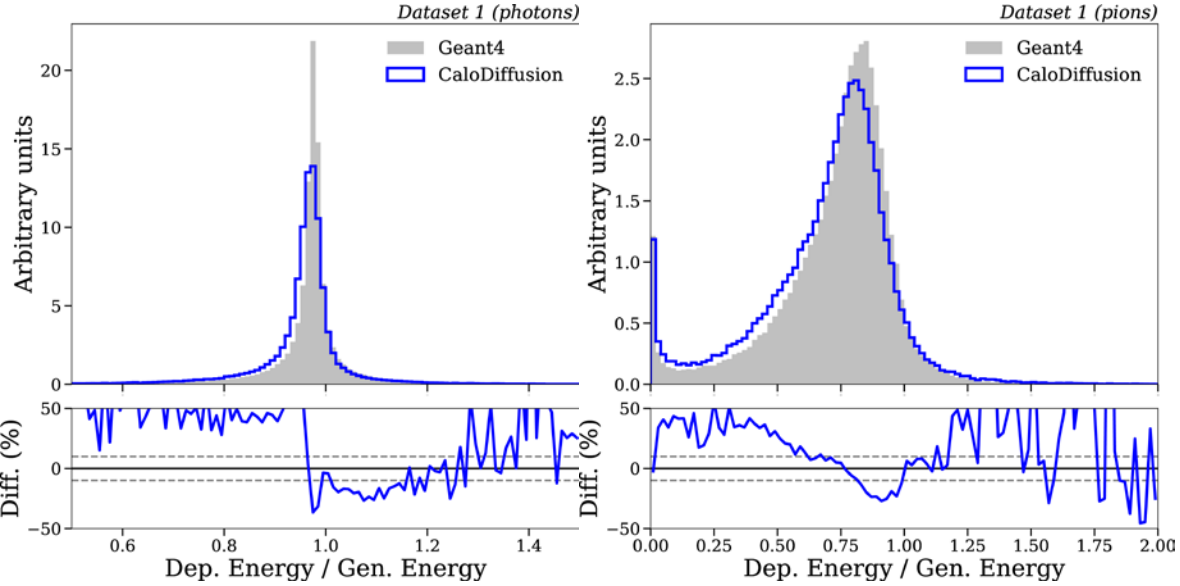
- Excellent modeling for photon showers
- Some mismodeling of low-energy pions
  - Could be resolved by dedicated training/conditioning
  - No significant impact on shower shape variables

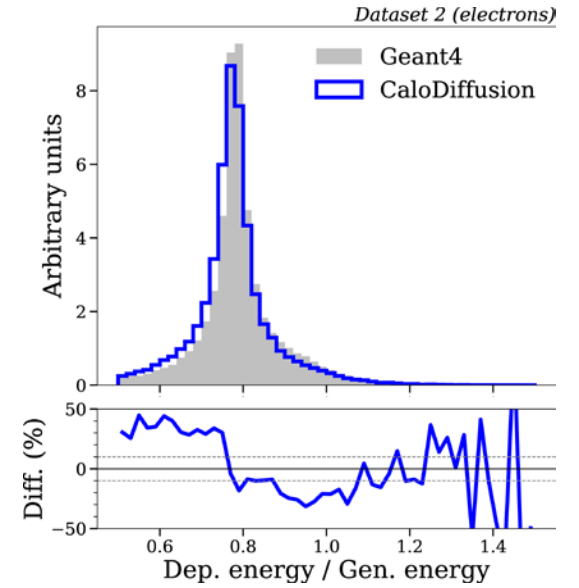
CaloDiffusion: CaloChallenge Datasets 2 & 3

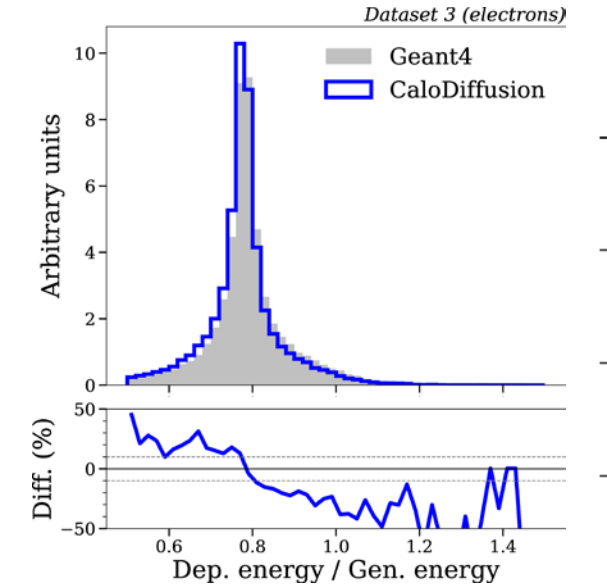


- Very good agreement in shower shapes and physically important quantities
- So far, have only shown 1D comparisons
- Next: further and higher-dimensional quantification

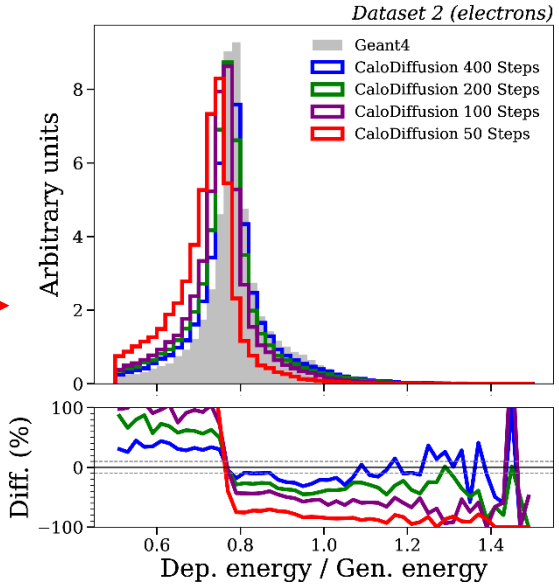
CaloDiffusion: Areas for Improvement







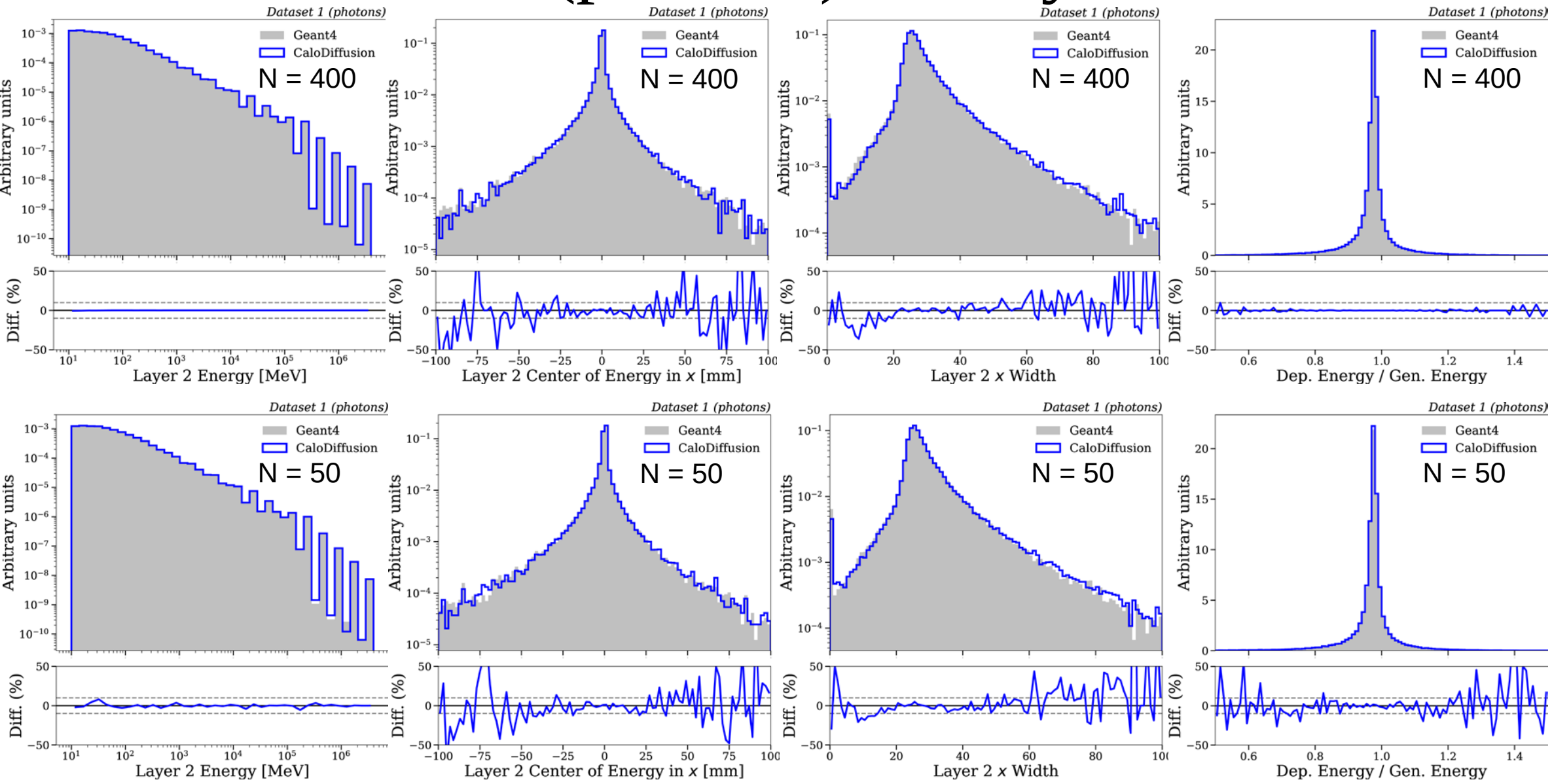
- Deficit in total energy modeling
- Need 400 diffusion steps to get acceptable quality
  - Still faster than Geant4 (~100s) w/ batching on GPU
- Fewer steps:
  - Linear speed improvement
  - But even less accurate in this quantity →



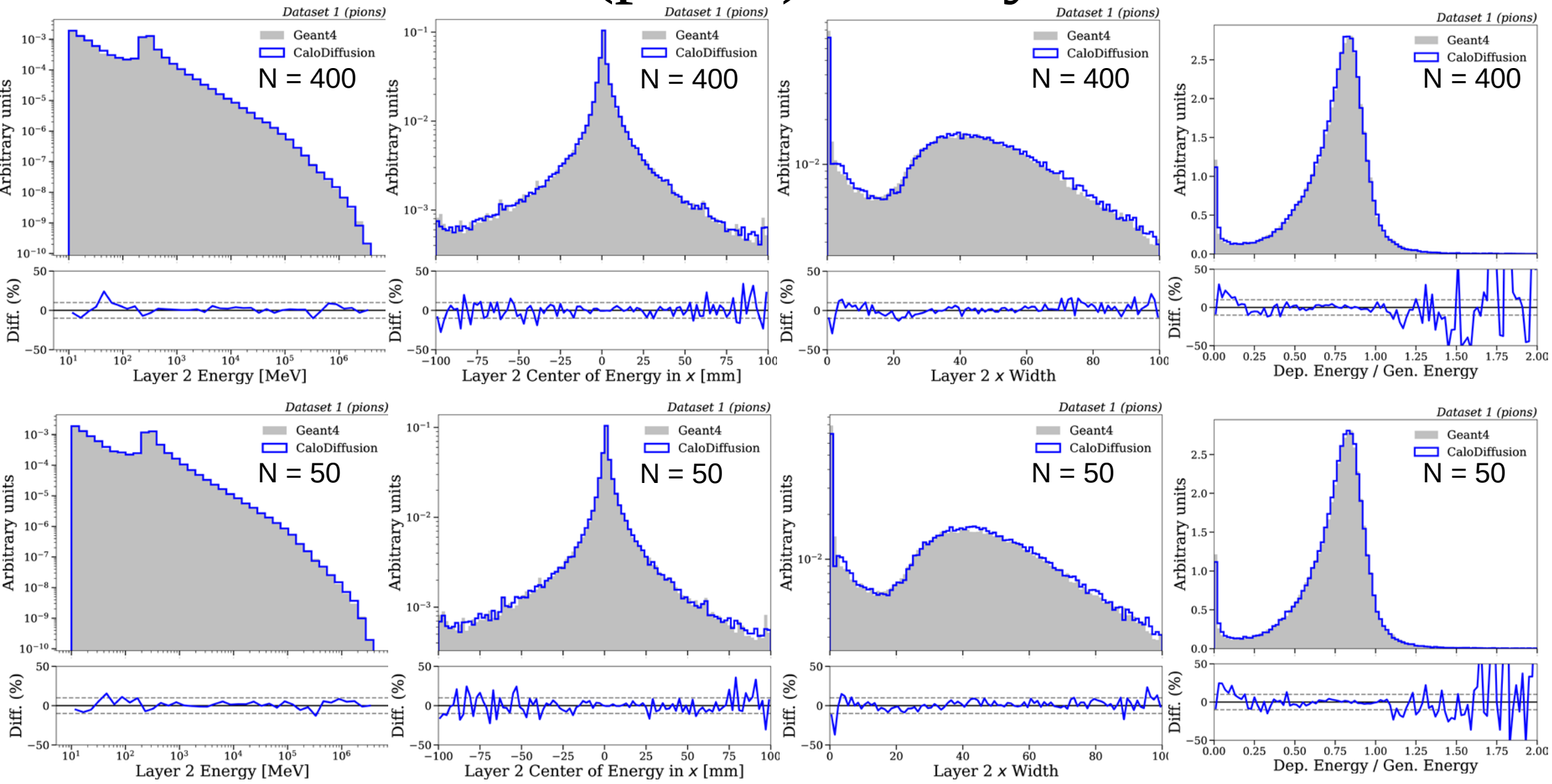
Dataset	Batch Size	Time/Shower [s] CPU	GPU
1 (photons) (368 voxels)	1	9.4	6.3
	10	2.0	0.6
	100	1.0	0.1
1 (pions) (533 voxels)	1	9.8	6.4
	10	2.0	0.6
	100	1.0	0.1
2 (electrons) (6.5K voxels)	1	14.8	6.2
	10	4.6	0.6
	100	4.0	0.2
3 (electrons) (40.5K voxels)	1	52.7	7.1
	10	44.1	2.6
	100	-	2.0

Num. Steps	Classifier AUC (low / high)	FPD	E Ratio Sep. Power
400	0.56 / 0.55	0.043(1)	0.011
200	0.61 / 0.56	0.046(1)	0.036
100	0.69 / 0.59	0.065(3)	0.079
50	0.83 / 0.67	0.110(4)	0.251

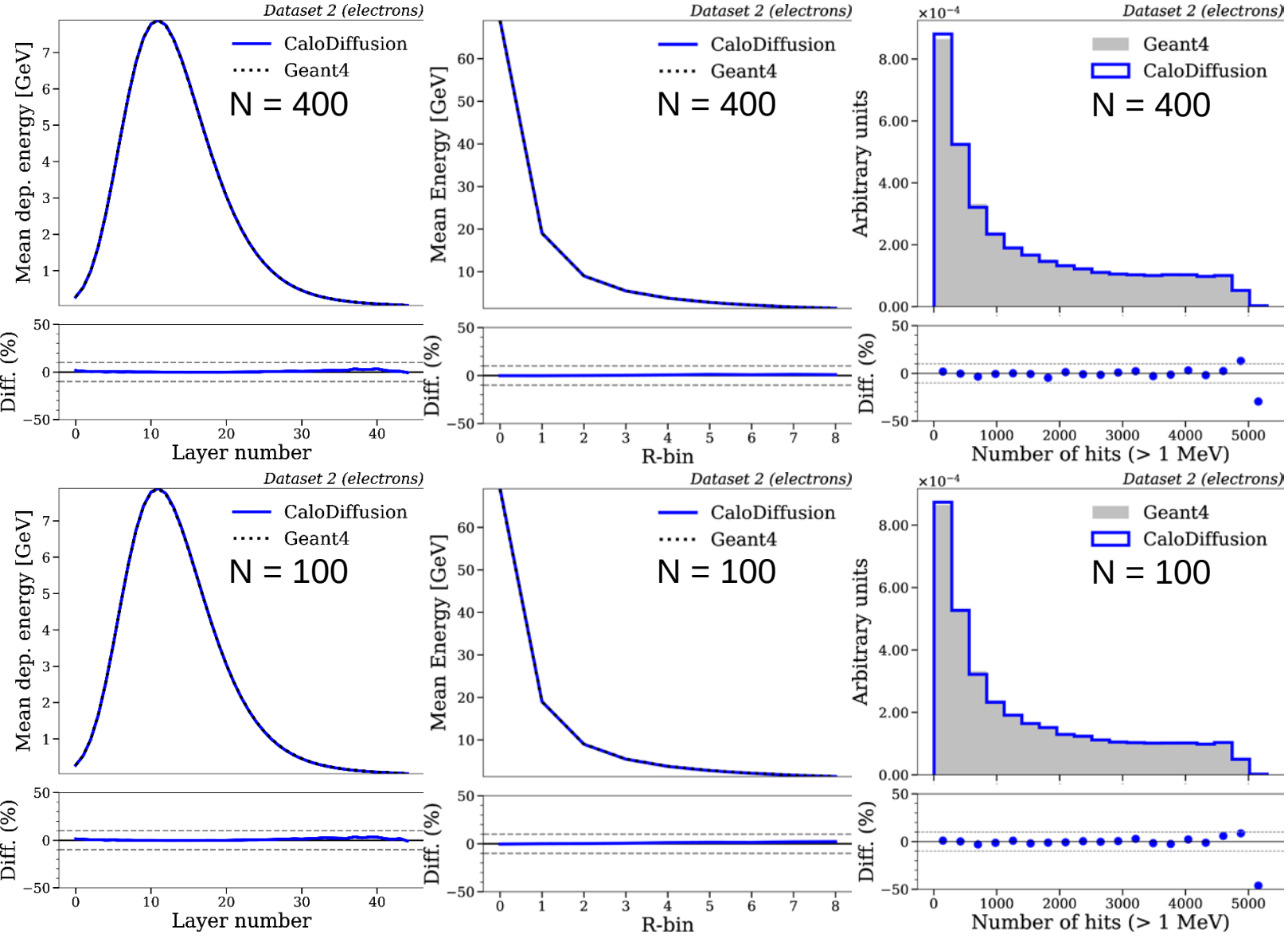
Dataset 1 (photons) w/ LayerDiffusion



Dataset 1 (pions) w/ LayerDiffusion



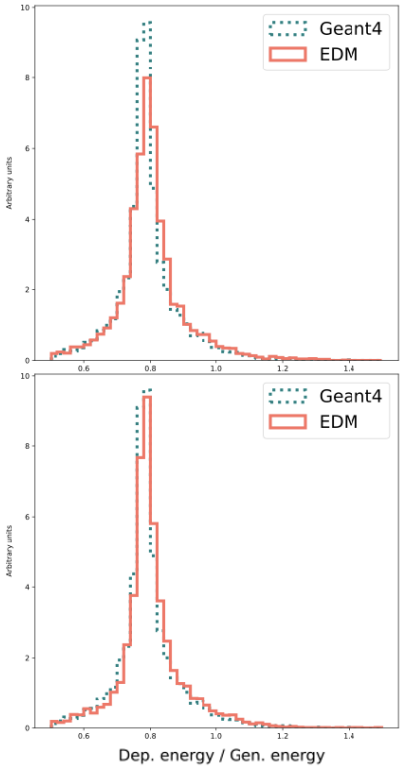
Dataset 2 w/ LayerDiffusion



- Virtually indistinguishable for 4× fewer diffusion steps
- Improved agreement vs. original CaloDiffusion

Model	AUC (low / high)	FPD	KPD ($\times 10^3$)	E Ratio Sep. Power
Old (N = 400)	0.56 / 0.56	0.043	0.1	0.011
New (N = 400)	0.54 / 0.58	0.045	0.05	0.0017
New (N = 100)	0.54 / 0.60	0.076	0.3	0.0017

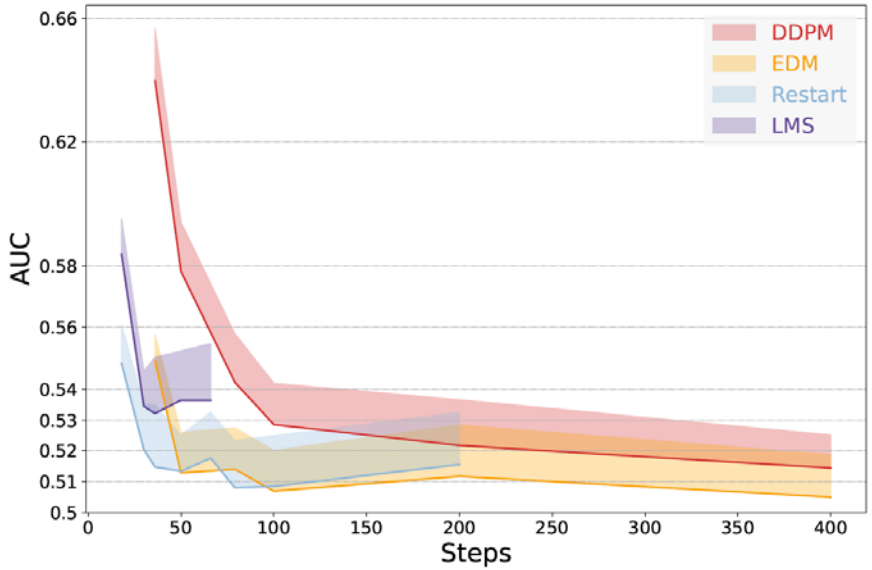
Choose Your Diffusion



default (unit) weights

min. signal to noise ratio weighting

Training: improved modeling from better weighting procedure

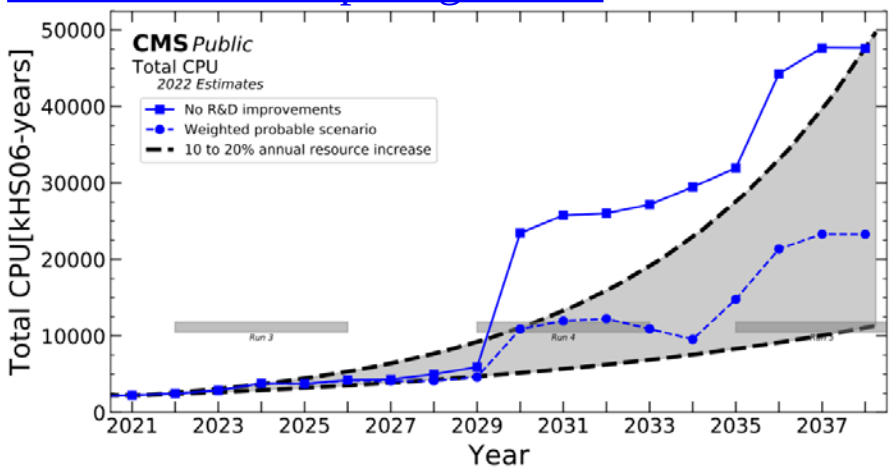


Inference: newer/smarter samplers for ODE/SDE solving can provide O(10×) speedup!

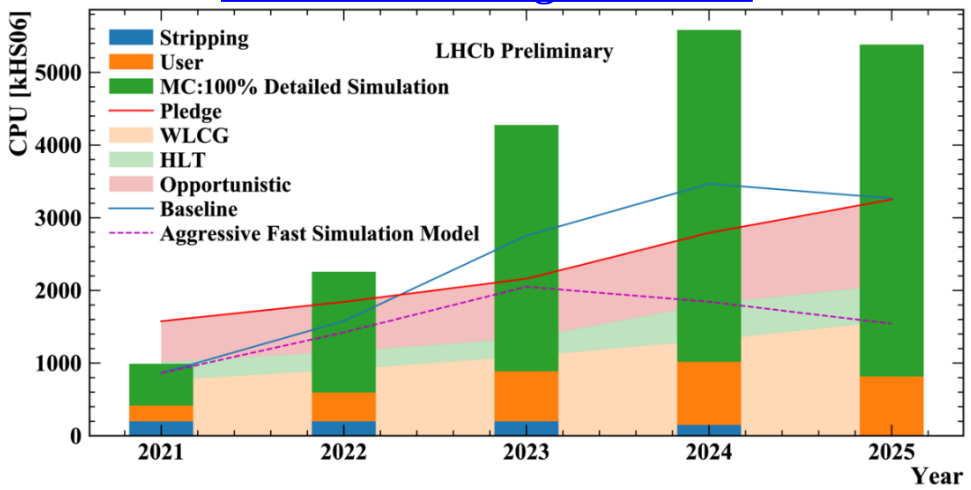
arXiv:2401.13162

LHC Computing Projections

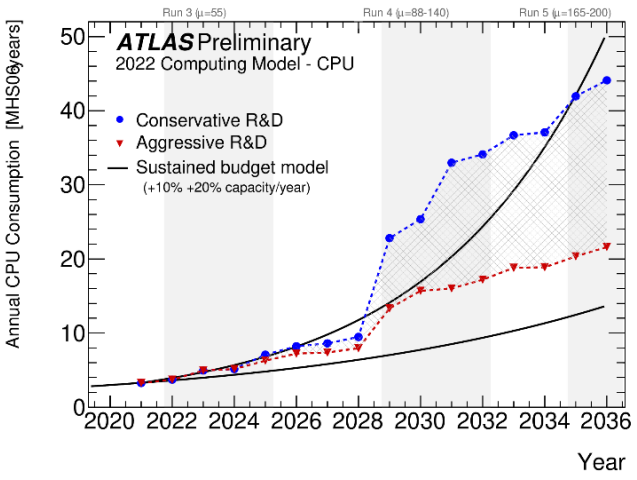
[CMSOfflineComputingResults]

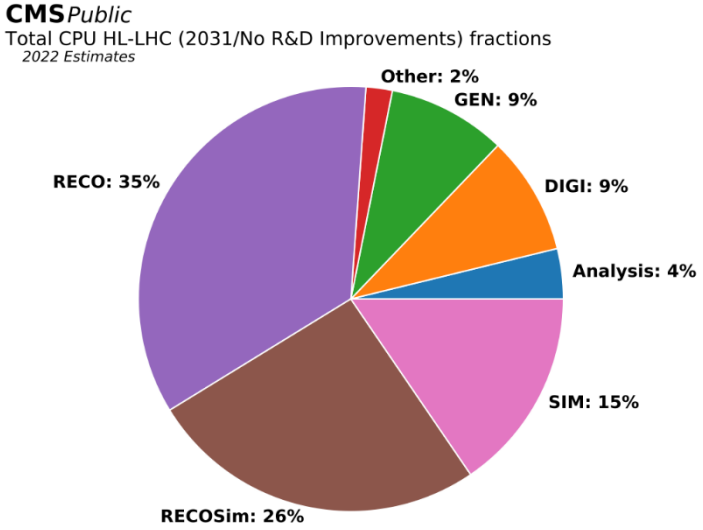


[LHCb CPU Usage Forecast]

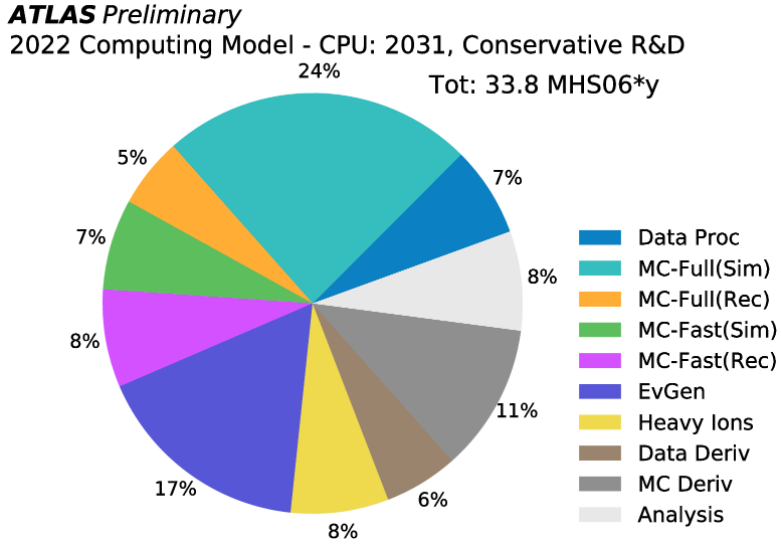


[CERN-LHCC-2022-005]

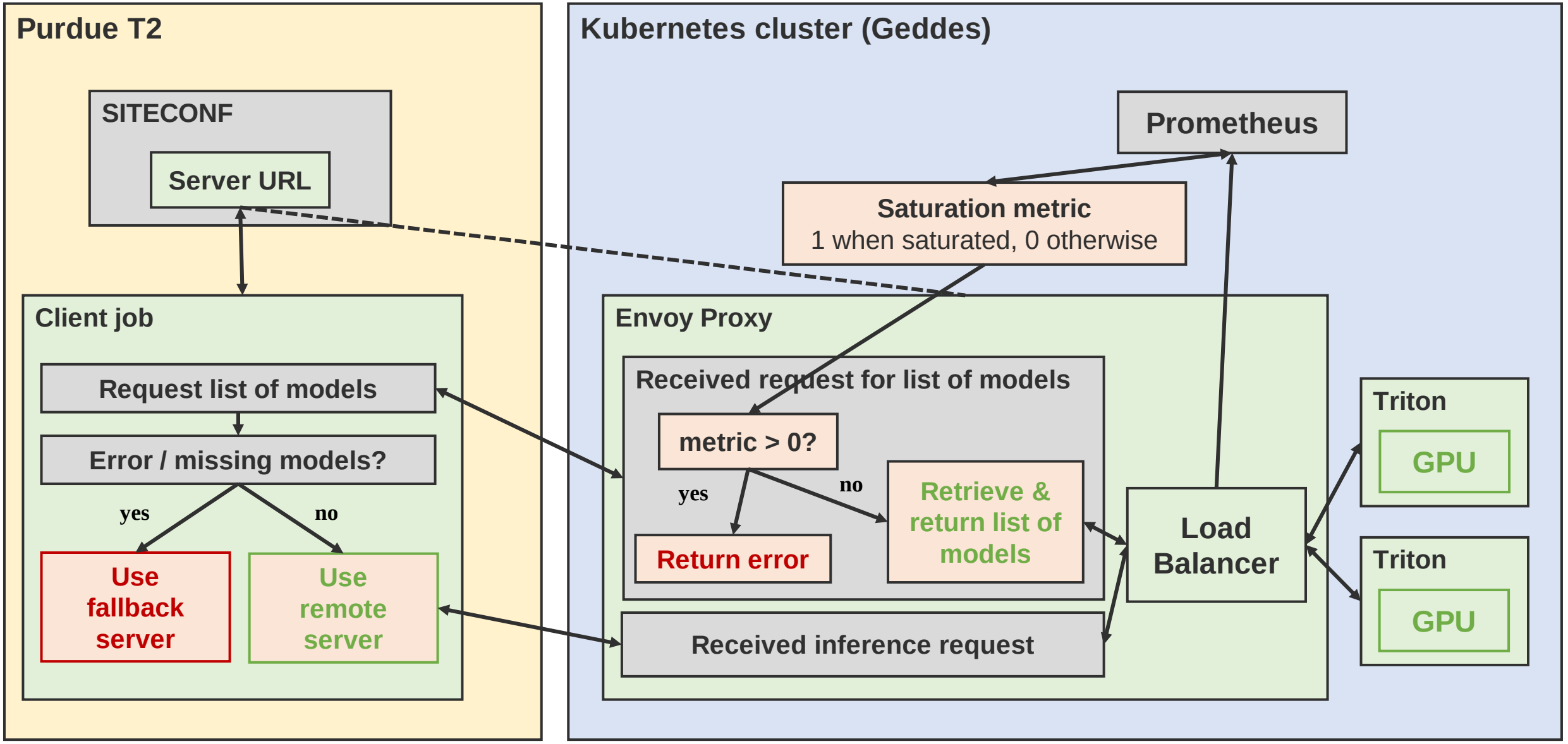




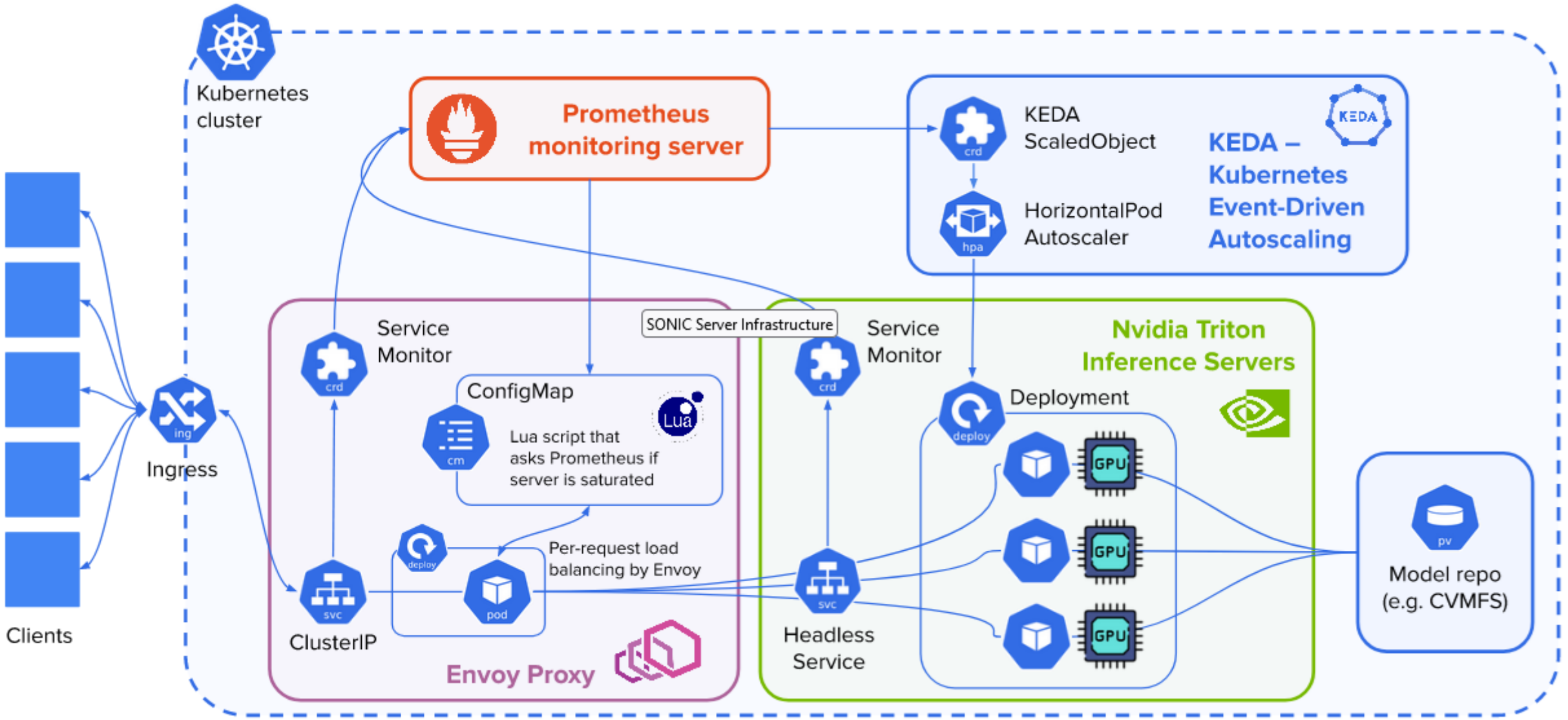
- Run 2: (full) simulation used ~40% (plurality) of grid computing resources for CMS & ATLAS [arXiv:1803.04165]
  - 70% for LHCb! [LHCb-PUB-2022-010]
- Run 4+: limit to ~10–20%, while simulating:
  - Complex detector upgrades
    - e.g. CMS High Granularity Calorimeter
  - More precise physics models
  - More events to reduce statistical uncertainty



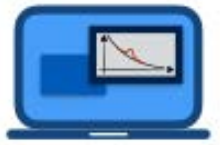
SONIC Server Architecture



SuperSONIC Architecture



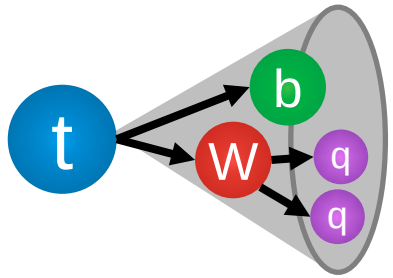
Classification



arXiv:2202.03772

	All classes		$H \to b\bar{b}$	$H \to c\bar{c}$	$H \to gg$	$H \to 4q$	$H \to \ell\nu qq'$	$t \to bqq'$	$t \to b\ell\nu$	$W \to qq'$	$Z \to q\bar{q}$
	Accuracy	AUC	$\text{Rej}_{50\%}$	$\text{Rej}_{50\%}$	$\text{Rej}_{50\%}$	$\text{Rej}_{50\%}$	$\text{Rej}_{99\%}$	$\text{Rej}_{50\%}$	$\text{Rej}_{99.5\%}$	$\text{Rej}_{50\%}$	$\text{Rej}_{50\%}$
PFN	0.772	0.9714	2924	841	75	198	265	797	721	189	159
P-CNN	0.809	0.9789	4890	1276	88	474	947	2907	2304	241	204
ParticleNet	0.844	0.9849	7634	2475	104	954	3339	10526	11173	347	283
ParT	**0.861**	**0.9877**	**10638**	**4149**	**123**	**1864**	**5479**	**32787**	**15873**	**543**	**402**
ParT (plain)	0.849	0.9859	9569	2911	112	1185	3868	17699	12987	384	311

- ML4Jets 2018: comparison studies eventually led to the Greatest Of All Taggers (GOAT)
  - Performance slightly exceeding ParticleNet
- Particle Transformer (ParT) is a *massive step forward*
  - Many more parameters, but fewer operations (→ faster!)
  - Uses pairwise features from 4-vectors → *domain knowledge*
- Other transformers like GN2X also being explored: $H \to b\bar{b}$ $\text{Rej}_{50\%} = 300$



Architecture	Accuracy	AUC	$1/\epsilon_B$	# Params
$\text{LorentzNet}_{n_{\text{hidden}}=3}$	0.9056	0.9623	136	120
$\text{nPELICAN}_{C_{\text{hidden}}=10}$	0.9208	0.9747	332	101
$\text{nPELICAN}_{C_{\text{hidden}}=3}$	0.9179	0.9722	272	31
$\text{nPELICAN}_{C_{\text{hidden}}=2}$	0.9169	0.9716	255	21
$\text{nPELICAN}_{C_{\text{hidden}}=1}$	0.8951	0.9480	70	11

arXiv:2310.16121

- Opposite side of the spectrum: PELICAN achieves competitive performance with only O(100) parameters!
  - Incorporates even more domain knowledge
  - $n_{\text{param}} = 10C_{\text{hidden}} + 1$

Foundation Models

- A growing trend in industry:
  - LLMs: finetuning or simply tokenizing additional data
    - Apply already-learned relationships to new information
  - Image generation: textual inversion, low-rank adaptation
- Back to ParT: not just >2M parameters, but trained on *100M jets*
  - *2 orders of magnitude* higher than previous datasets
  - Fine-tuned version performs *better* on those previous datasets than fully retrained version!
  - ➢ Learns *generalized physics* that can apply to many datasets
- Maybe the *first glimmer* of foundation models for physics
  - Build on techniques mentioned here (and others) to improve generalization and physics learning
  - Diffusion models could be foundation for generative tasks…

arXiv:2202.03772

	Accuracy	AUC	$\text{Rej}_{50\%}$	$\text{Rej}_{30\%}$
P-CNN	0.930	0.9803	201 ± 4	759 ± 24
PFN	—	0.9819	247 ± 3	888 ± 17
ParticleNet	0.940	0.9858	397 ± 7	1615 ± 93
JEDI-net (w/ $\sum O$)	0.930	0.9807	—	774.6
PCT	0.940	0.9855	392 ± 7	1533 ± 101
LGN	0.929	0.964	—	435 ± 95
rPCN	—	0.9845	364 ± 9	1642 ± 93
LorentzNet	0.942	0.9868	498 ± 18	2195 ± 173
ParT	0.940	0.9858	413 ± 16	1602 ± 81
ParticleNet-f.t.	0.942	0.9866	487 ± 9	1771 ± 80
ParT-f.t.	**0.944**	**0.9877**	$\mathbf{691 \pm 15}$	$\mathbf{2766 \pm 130}$